Scientific Excellence • Resource Protection & Conservation • Benefits for Canadians
Excellence scientifique • Protection et conservation des ressources • Bénéfices aux Canadiens

System Documentation for the Mark Recovery Program Database Second Edition

M. A. Holmes and J. M. Hamer

Biological Sciences Branch
Department of Fisheries and Oceans
Pacific Biological Station
Nanaimo, British Columbia V9R 5K6

1992

Canadian Technical Report of Fisheries and Aquatic Sciences 1858

Fisheries and Oceans Pêches et Océans

Canada

Canadian Technical Report of Fisheries and Aquatic Sciences

Technical reports contain scientific and technical information that contributes to existing knowledge but which is not normally appropriate for primary literature. Technical reports are directed primarily toward a worldwide audience and have an international distribution. No restriction is placed on subject matter and the series reflects the broad interests and policies of the Department of Fisheries and Oceans, namely, fisheries and aquatic sciences.

Technical reports may be cited as full publications. The correct citation appears above the abstract of each report. Each report is abstracted in *Aquatic Sciences and Fisheries Abstracts* and indexed in the Department's annual index to scientific and technical publications.

Numbers 1–456 in this series were issued as Technical Reports of the Fisheries Research Board of Canada. Numbers 457–714 were issued as Department of the Environment, Fisheries and Marine Service, Research and Development Directorate Technical Reports. Numbers 715–924 were issued as Department of Fisheries and the Environment, Fisheries and Marine Service Technical Reports. The current series name was changed with report number 925.

Technical reports are produced regionally but are numbered nationally. Requests for individual reports will be filled by the issuing establishment listed on the front cover and title page. Out-of-stock reports will be supplied for a fee by commercial agents.

Rapport technique canadien des sciences halieutiques et aquatiques

Les rapports techniques contiennent des renseignements scientifiques et techniques qui constituent une contribution aux connaissances actuelles, mais qui ne sont pas normalement appropriés pour la publication dans un journal scientifique. Les rapports techniques sont destinés essentiellement à un public international et ils sont distribués à cet échelon. Il n'y a aucune restriction quant au sujet; de fait, la série reflète la vaste gamme des intérêts et des politiques du ministère des Pêches et des Océans, c'est-à-dire les sciences halieutiques et aquatiques.

Les rapports techniques peuvent être cités comme des publications complètes. Le titre exact paraît au-dessus du résumé de chaque rapport. Les rapports techniques sont résumés dans la revue *Résumés des sciences aquatiques et halieutiques*, et ils sont classés dans l'index annual des publications scientifiques et techniques du Ministère.

Les numéros 1 à 456 de cette série ont été publiés à titre de rapports techniques de l'Office des recherches sur les pêcheries du Canada. Les numéros 457 à 714 sont parus à titre de rapports techniques de la Direction générale de la recherche et du développement, Service des pêches et de la mer, ministère de l'Environnement. Les numéros 715 à 924 ont été publiés à titre de rapports techniques du Service des pêches et de la mer, ministère des Pêches et de l'Environnement. Le nom actuel de la série a été établi lors de la parution du numéro 925.

Les rapports techniques sont produits à l'échelon régional, mais numérotés à l'échelon national. Les demandes de rapports seront satisfaites par l'établissement auteur dont le nom figure sur la couverture et la page du titre. Les rapports épuisés seront fournis contre rétribution par des agents commerciaux.

Canadian Technical Report of
Fisheries and Aquatic Sciences 1858

1992

SYSTEM DOCUMENTATION FOR THE MARK RECOVERY PROGRAM DATABASE

SECOND EDITION

by

M. A. Holmes and J. M. Hamer

Biological Sciences Branch
Department of Fisheries and Oceans
Pacific Biological Station
Nanaimo, British Columbia V9R 5K6

93-00917/

(c) Minister of Supply and Services Canada 1992

Cat. No. Fs 97-6/1858E ISSN 0706-6457

Correct citation for this publication:

Holmes, M. A. and J. M. Hamer. 1992. System documentation for the Mark Recovery Program Database. Second Edition. Can. Tech. Rep. Fish. Aquat. Sci. 1858: 94 p.

TABLE OF CONTENTS

ABSTRACT v

I. INTRODUCTION AND BACKGROUND 1
SYSTEM DESIGN 1
DATA STRUCTURE 2

UPDATES 2

II. TECHNICAL DESCRIPTIONS 3
COMMON BLOCKS 3
Cwt Release Common Block 3
Finclip Release Common Block 4
Commercial Recovery Common Block 6
SOURCE CODE MODULES 8
Commercial Sample Source Code 10
Escapement Sample Source Code 10
Random Recovery Common Block 10
Sport Recovery Matrix 11
Sport Mark Incidence Counts 11
Catch Sample Area Matrix 12
SUBROUTINE TABLES 13
Table Type Chart 14
LIBRARY TABLES 14
Catch Region Name Table 17
Program Type Table 18
Production Area Table 19
Sport Catch Region Table 19
SUBROUTINES 21
Subroutine Descriptions 22
Subroutine Hierarchy 22
Subroutines and Restricted Numbers 24
CWTAJIM 26
CWTCALWEK 27
CWTCHAIN 28
CWTCHAINS 29
CWTCMSTCR 31
CWTCOLOUR 32
CWTCRWK 34
CWTCSGET 35
CWTDW3D4 36
CWTESGET 37
CWTGRNAM 38
CWTHISITE 39
CWTLISTREL 40
CWTMI1 41
CWTMI2 42
CWTMMW 43
CWTMMW1 44
45

CWTNONGET 47
CWTNONOPN 48
CWTNONTAG 49
CWTOPCHN 50
CWTOPMST 51
CWTPAREA 52
CWTPROGTYP 54
CWTREARTYP 55
CWTRECRD 56
CWTREGNAM 58
CWTRELEASE 59
CWTRELGET 61
CWTRELRD 62
CWTSEXMAT 63
CWTSITGET 64
CWTSML 65
CWTSPORTLOC 66
CWTSPSTCR 67
CWTSTAGE 68
CWTSTATCS 69
CWTSTKTYP 70
CWTSUM1 71
CWTVTP 72
FINCHAINS 73
FINOPMST 74
FINRELGET 75
FINRELRD 76
FINTOCWT 77
MICROTRAN 78
REGIONGR 79
RELSTAGE 80
RUNTYPE 81
SAMLOC 82
SPECIES 83
STATSUB 84
STUDYTYPE 86

APPENDIX A 87
APPENDIX B 88
APPENDIX C 90
APPENDIX D 91
APPENDIX E 92
REFERENCES 94

ABSTRACT

Holmes, M. A. and J. M. Hamer. 1992. System documentation for the Mark Recovery Program Database. Second Edition. Can. Tech. Rep. Fish. Aquat. Sci. 1858: 94 p.

The Biological Sciences Branch in Nanaimo, B.C. at the Pacific Biological Station, maintains the Salmonid Catch Sampling and Mark Recovery Program database on the VAX computer for the Pacific Region of the Canadian Department of Fisheries and Oceans. The database contains coded-wire tag releases (Jefferts et al. 1963) from fishery agencies in California and up the coast to British Columbia and through to S.E. Alaska. Recoveries are available of coded-wire tag salmonids in all the Pacific coastal fisheries since 1975, plus there is sampling information related to recovery effort and marked escapement catch samples. As well as MRP sampling data, catch data from the BSB database on commercial salmon catches (Wong 1983, Holmes 1991) and salmon sport catch and sampling data provided by Fisheries Branch are present. Information is accessible via menu driven report generators for commonly requested data, through a spreadsheet program, and/or via user written programs employing the MRP subroutine library. This report updates and supercedes an earlier technical report on the same topic (Holmes, M. and J. M. Hamer. 1988. System documentation for the Mark Recovery Program. Can. Tech. Rep. Fish. Aquat. Sci. 1601: 103 p.)

Key words: coastal fisheries, coded-wire tag, mark recovery, sampling, salmonid, salmon database, salmon releases, salmon recoveries, salmon sport

RÉSUMÉ

Holmes, M. A. and J. M. Hamer. 1992. System documentation for the Mark Recovery Program Database. Second Edition. Can. Tech. Rep. Fish. Aquat. Sci. 1858: 94 p.

C'est à la Direction des sciences biologiques de la Station biologique du Pacifique (Nanaimo, C.-B.), qu'est conservée sur ordinateur VAX la base de données du Programme de reprise de poissons étiquetés et d'échantillonnage des salmonidés (région du Pacifique) du ministère canadien des Pêches et des Océans. Cette base renferme des données sur les poissons portant des micromarques magnétisées codées (Jefferts et al. 1963) libérés par des organismes de pêche de la côte ouest (depuis la Californie jusqu'au sud-est de l'Alaska). Ces données sont disponibles à partir de 1975 pour toutes les pêches côtières du Pacifique; il y a en outre de l'information sur l'effort de reprise et sur les poissons marqués présents dans les échappées. En plus des données d'échantillonnage en vertu du Programme de reprise de poissons étiquetés, la base comporte des données de capture provenant de la base de données de la Direction des Sciences biologiques sur les captures commerciales de saumon (Wong, 1983; Holmes, 1991) et des données d'échantillonnage et de pêche sportive de saumon fournies par la direction des pêches. L'information est disponible par l'entremise de générateurs de rapports pilotés par menus (dans le cas des données demandées fréquemment), d'un tableur et/ou de programmes écrits pour l'utilisateur employant une bibliothèque de sous-programmes du Programme de reprise de poissons étiquetés. Le présent rapport met à jour et remplace un rapport technique précédent portant sur le même sujet (Holmes, M. et J. M. Hamer. 1988. System documentation for the Mark Recovery Program. Can. Tech. Rep. Fish. Aquat. Sci. 1601: 103 p.)

Mots-clés: pêches côtières, micromarque magnétisée codée, récupération des marques, échantillonnage, salmonidés, base de données sur les saumons, saumons libérés, échappées, pêche sportive (saumon)

I. INTRODUCTION AND BACKGROUND

This report updates the first technical report on Mark Recovery Database documentation - #1601. Since it's publication, international data exchange and the agreed upon data format changes have required that common blocks and some subroutines be updated and/or created.

The system described in this report is resident on the VAX computer at the Pacific Biological Station in Nanaimo, B.C. and is available to all registered users. Coded-wire tag sampling as well as the keypunching of commercial and sport data are managed under Fisheries Branch contracts. The Biological Sciences Branch (BSB) is responsible for marked escapement data entry and all other historical database management. System architecture was designed by J.M. Hamer and L.A.Lapi while a number of interested user groups from Fisheries and the public sector researched, developed and implemented the database (Kuhn, 1988). The Salmon Database group handles database maintenance and updates.

This second document updates the first document and reflects the additional subroutines and updated common blocks brought about by the change in specifications for reporting salmonid production and CWT data. As a member of the Pacific Salmon Commission (PSC), Canada is obligated to exchange marked salmon information for release, catch sample and recovery data with participating American agencies. In February 1990, version 2.0 was approved (Specifications for Reporting Salmonid Production and CWT Data, PSC Format Version 2.0, 22 February 1990). This updated version of the system documentation reflects these changes.

The intent of this report is to provide programming documentation for subroutines, tables and common blocks that are needed for accessing the MRP database, which is written in the FORTRAN programming language. Experienced programmers will use this documentation to obtain listings of necessary common blocks and subroutines for their own access to mark recovery data. Section I outlines past systems, present system design, data flow, pointers and reports. Section II documents common blocks, tables and subroutines. The appendices includes among other things, examples of release and recovery program shells.

SYSTEM DESIGN

Release data are present from 1967 while commercial and marked escapement samples are present from 1975. Data access is done in the FORTRAN programming language, either by chaining; binary searches; with pointers; or/and with backward and forward references. Users do not need detailed knowledge of the data files to use the subroutines, common blocks and tables. However, some knowledge of data structure can be used to make the accessing

programs run more efficiently. Appendices C, D and E list example programs for reading various files using subroutines and common blocks.

DATA STRUCTURE

Release data are maintained in two files, one file for CWT marked fish and one file for fin-marked fish. Either release file can be accessed sequentially or directly by tag code.

Separate files are maintained by year for catch/sample, escapement/sample and recovery data. Recovery data is accessed either 1) sequentially, 2) directly by catch region and time period, or 3) by tag code.

UPDATES

System updates are done periodically to preliminary data to incorporate new data or to process revisions to existing data. Canadian release data are entered onto the database by the Mark Recovery technician in Fisheries Services, as information on each release is received from the user. American release information is loaded from tapes sent by the Pacific States Marine Fisheries Commission (PSMFC) in Portland Oregon.

II. TECHNICAL DESCRIPTIONS

COMMON BLOCKS

The Mark Recovery database system uses three labelled common blocks to pass the large number of variables associated with a release or a recovery without having to resort to large argument lists. The common blocks are:

[Cwtsys.Release]RELRECCOM.FOR - CWT release master common block

[Finclip.Release]FINRELCOM.FOR - FINCLIP release master common block

[Cwtsys.Recovered]RECTAGCOM.FOR - CWT recovery master common block

The FORTRAN **INCLUDE** verb is used to tell the FORTRAN compiler to insert a file into the source code at the point where it sees the include directive. The file to be included can contain anything from comments, variable definitions, common blocks, or even executable statements or subroutines.

In the MRP system, **INCLUDE** handles two routines:

1. Common blocks for releases and recoveries. These modules contain the variable declarations and common block definitions. The library routines communicate with user code via these block names. The blocks are included in both the library routine and in the user code.

2. Variable declarations only i.e. STATCSCOM.FOR (the "COM" is a bit misleading but does make the file naming consistent). These files contain only the declaration of a variable. There is no common block name i.e. no link between these modules and CWTLIB (CWT library) routines. When a user calls one of these routines, they must pass the arguments in the CALL statement i.e. CALL CWTCSGET(CR,YR,CSREC,OPT,ERR). Please see Appendix B for statweek definition.

Cwt Release Common Block

```
C_____________________________________________________
C  COMMON BLOCKS FOR A RELEASE RECORD  VERSION 2
C  JAN91
C
      INTEGER*4 IVAR(40)
      REAL*4    RVAR(10)
      INTEGER*4 RELNEST(10)
      INTEGER*4 YRADDR(10)
      BYTE      YRSBAC(10)
C
C
      COMMON /RELRECNBLK/ IVAR,RVAR,RELNEST,YRADDR,YRSBAC
C
      CHARACTER AGENCY*4,COMMENT*80,STAGECOD*4
      CHARACTER TAG*12,DELSW*1,STUDY*1,COMPLT*1
      CHARACTER MANDFLAG*15
      CHARACTER RELREAR*1,RELSTK*1,RELEXID*4,RELMETHOD*1
      CHARACTER RELSURV*1,RELPROG*3,RELSPARE*21
      CHARACTER RELSW(10)*1
C
      COMMON /RELRECCBLK/  TAG,DELSW,STUDY,COMPLT,
     .AGENCY,COMMENT,STAGECOD,MANDFLAG,
     .RELREAR,RELSTK,RELEXID,RELMETHOD,RELSURV,RELPROG,
     .RELSPARE,RELSW
C
C     IVARS
C
C  01 HART
C  02 BROODYEAR
C  03 RUN
C  04 DAY FIRST RELEASE
C  05 MON
C  06 YEAR
C  07 DAY LAST REL
C  08 MON
C  09 YR
C     IF DD-MM-YY OF DATE LAST REL ALL 0 AND MANDFLAG(8:8)=Y
C     THEN RELEASE HAD 100% MORTALITY (SEE ALSO "RELSURV")
C  10 NUM TAGGED
C  11 AD_ONLY
C  12 UNCLIPPED
C  13 TOTAL
C  14 DAYS HELD  "-1" =  UNKNOWN.
C  15 SIZE CODE 1- FSH/LB 2- GR/FISH 3-LEN CM 0-NO DATA ++
C  16 DAY LAST UPDATE
C  17 MON LAST UPDATE
C  18 YR  LAST UPDATE
C  19 HATCHERY NUM (ADDR IN SITES FILE)
C  20 REL SITE "
C  21 STOCK    "
```

```
C  22 COORDINATOR CODE
C       ON CHAINS, CHAIN=0  MEANS YOU GOT THE LAST ONE "CURRENT"
C  23   CHAIN TO LAST RECORD SAME HAT
C  24     "       "     "       "  RELSITE
C  25     "                        STOCK
C  26   HIGHEST REPLICATE CODE. (0="PLAIN TAG")
C  27   LENGTH (MM) (SEE ALSO RVAR(05))
C  28   RESERVED FOR MRP SYSTEM USAGE. (AUTOLOAD SW)
C  29   FOR SEQUENTIAL TAGS - STARTING DATA3
C  30   FOR SEQUENTIAL TAGS - STARTING DATA4
C  31   FOR SEQUENTIAL TAGS - ENDING DATA3
C  32   FOR SEQUENTIAL TAGS - ENDING DATA4
C  33   FOR SEQUENTIAL TAGS - LOWER BOUND SEQ NUMBER =0 ON NON-SEQ WIRE
C  34   FOR SEQUENTIAL TAGS - UPPER BOUND SEQ NUMBER =0 ON NON-SEQ WIRE
C  35   TAGLOSS SAMPLE SIZE   -1 = UNKNOWN
C  36-40 SYSTEM USE
C
C   RVARS
C  01 %TAGLOSS
C  02 SIZE  SEE IVAR(15) ++
C  03 BIOSTD (EXPECTED SURVIVAL)"-1" = UNKNOWN !NO LONGER CARRIED
C  04 WT (GM/FISH)
C  05 LN (MM) (SEE ALSO IVAR(27))
C  06-10 SPARES
C
C  ++ - "FOR COMPATIBILITY WITH OLD PROGS".NEW CODE
C        SHOULD USE IVARS 27 OR RVAR 5 FOR LN, RVAR 4 FOR WT.
C
C    OTHER STUFF
C
C    RELSW(01)=Y = IS A SEQUENTIAL WIRE RELEASE
C    RELSW(02)= C = CDN RELEASE    = U = USA RELEASE
C    RELSURV =D MEANS RELEASE HAD 100% MORTALITY.
C               (IVARS 4 THRU 13 = 0)
C    STUDY = P,W,E,B,M = PROD,WILD,EXPERIMENTAL,BOTH P & E, MONITOR
C    RELREAR = H,W,F = HAT,WILD,FED WILD = REARING TYPE
C    RELSTK  = H,W,M = HAT,WILD,MIXED H & W STOCK = STOCK TYPE
C    RELMETHOD = B,C,P,W = BOOK,COUNTS,PETERSEN,WEIGHT = COUNTING METHOD
C
C    COMPLT =Y  RELEASE IS COMPLETE
C           =N  INCOMPLETE
C    DELSW  =N  IS VALID RECORD.(ALL RECORDS WILL BE "N")
C               (FOR COMPATIBILITY WITH "OLD PROGS")
C
C     COMPLETION LEVEL.
C     A "Y" IN THE CORRESPONDING POSITION OF MANDFLAG MEANS
C     THAT FIELD IS COMPLETE.(USE TO SEE WHY RELEASE INCOMPLETE)
C
C     1AGENCY,2HART,       3BY,   4STK,    5STUDY,6HATCHERY
C     7RELSIT,8DT(LASTREL),9NTAG,10ADONLY,11UNCLIP
C
C     YRSBAC - CONTAINS THE LAST TWO DIGITS OF A YEAR IN WHICH
```

```
C              THERE WAS A RETURN.
C              CORRESPONDING YRADDR CONTAINS START CHAIN ADDRESS
C              IN THE RECTAGSyy.DAF FILE.
C     RELNEST - ESTIMATED RECOVERIES CORRESPONDING TO YRSBAC !NOT IMPLEMENTED
C
C_____________________________________
```

Finclip Release Common Block

```
C_____________________________________
C  COMMON BLOCKS FOR A FINCLIP RELEASE RECORD  V2
C  JAN91
C  FINCLIP RTNS WERE "LIFTED" FROM CWT RTNS. NOTE THAT
C  BECAUSE OF THIS,SOME OF THE FIELDS IN FINCLIP RECORDS
C  ARE NOT MEANINGFUL AS DESCRIBED.
C  ALSO - NOTE THAT THE MAIN DIFFERENCES BETWEEN CWT
C  FIELD NAMES AND FINCLIP FIELD NAMES ARE SO:
C
C      CWT                  FINCLIP
C
C     IVARS                 JVARS
C     RVARS                 SVARS
C     YRADDR,YRSBAC         ZRADDR,ZRSBAC
C     ON CHARACTER FIELDS,THE FIRST CHARACTER OF THE CWT
C     CHARACTER VARIABLE IS REPLACE WITH AN "F" FOR FINCLIP.
C     EG   AGENCY   FGENCY  EXCEPT FLAG1 IS FFLAG1
C
C
      INTEGER*4 JVAR(40)
      REAL*4    SVAR(10)
      INTEGER*4 JELNEST(10) ! RECOV STUFF  NONE FOR CLIPS
      INTEGER*4 ZRADDR(10)  !  "
      BYTE      ZRSBAC(10)  !  "
C
C
      COMMON /FINRELNBLK/ JVAR,SVAR,JELNEST,ZRADDR,ZRSBAC
C
      CHARACTER FGENCY*4,FOMMENT*80,FTAGECOD*4
      CHARACTER FAG*12,FELSW*1,FTUDY*1,FOMPLT*1
      CHARACTER FANDFLAG*15
      CHARACTER FELREAR*1,FELSTK*1,FELEXID*4,FELMETHOD*1
      CHARACTER FELSURV*1,FELPROG*3,FELSPARE*21
      CHARACTER FELSWTCH(10)*1
C
      COMMON /FINRELCBLK/  FAG,FELSW,FTUDY,FOMPLT,
     .FGENCY,FOMMENT,FTAGECOD,FANDFLAG,
     .FELREAR,FELSTK,FELEXID,FELMETHOD,
     .FELSURV,FELPROG,FELSPARE,FELSWTCH
C
C     JVARS
C
```

```
C  01 HART
C  02 BROODYEAR
C  03 RUN
C  04 DAY FIRST RELEASE
C  05 MON
C  06 YEAR
C  07 DAY LAST REL
C  08 MON
C  09 YR
C     IF DD-MM-YY OF DATE LAST REL ALL 0 AND FANDFLAG(8:8)=Y
C     THEN RELEASE HAD 100% MORTALITY
C  10 NUM TAGGED
C  11 AD_ONLY
C  12 UNCLIPPED
C  13 TOTAL
C  14 DAYS HELD  "-1" =  UNKNOWN.
C  15 SIZE CODE 1- FSH/LB 2- GR/FISH 3-LEN CM 0-NO DATA
C     USERS SHOULD USE SVAR(4) AND SVAR(5) FOR WT/LN
C  16 DAY LAST UPDATE
C  17 MON LAST UPDATE
C  18 YR  LAST UPDATE
C  19 HATCHERY NUM (ADDR IN SITES FILE)
C  20 REL SITE "
C  21 STOCK    "
C  22 COORDINATOR CODE
C      ON CHAINS, CHAIN=0  MEANS YOU GOT THE LAST ONE "CURRENT"
C  23  CHAIN TO LAST RECORD SAME HAT
C  24    "       "     "      "  RELSITE
C  25    "                       STOCK
C  26  NOT USED ON FINCLIP
C  27  LN  (MM)  ( SEE ALSO SVAR(05) )
C  28  RESERVED FOR SYSTEM USE. (AUTLOAD SW)
C  29-35 RESERVED FOR SYSTEM USE (CWT)
C  36-40 SPARES
C
C
C   SVARS
C  01 %TAGLOSS
C  02 SIZE  SEE JVAR(15)
C  03 BIOSTD (EXPECTED SURVIVAL)  .LT. 0.0 = UNKNOWN
C  04 WT (GM/FISH)
C  05 LN (MM)  ( SEE ALSO JVAR(27) )
C  06-10 SPARE
C  SPARES
C
C    OTHER STUFF
C
C    FELSW   FOR COMPAT WITH OLD SYS.  WILL ALWAYS BE "N"
C
C    FELSWTCH(01)  UNUSED
C    FELSWTCH(02) = C = CDN RELEASE   = U = USA RELEASE
C
```

```
C    FOMPLT =Y  RELEASE IS COMPLETE
C           =N  INCOMPLETE
C
C     COMPLETION LEVEL.
C     A "Y" IN THE CORRESPONDING POSITION OF FANDFLAG MEANS
C     THAT FIELD IS COMPLETE.(USE TO SEE WHY RELEASE INCOMPLETE)
C
C     1AGENCY,2HART,3BY,4STK,5STUDY
C     6HAT,   7REL, 8RDL,9NTAG,10ADPERTL
C    11UNCLIP
C
C     ZRSBAC - CONTAINS THE LAST TWO DIGITS OF A YEAR IN WHICH
C              THERE WAS A RETURN.
C              CORRESPONDING ZRADDR CONTAINS START CHAIN ADDRESS
C              IN THE RECTAGSyy.DAF FILE.
C
C_____________________________________
```

Commercial Recovery Common Block

```
C-------------------------------------------------------------------------
C       COMMON BLOCK FOR A TAGRECOVERY
C       'CWT:[CWTSYS.RECOVERED]RECTAGCOM.FOR'
C                                                VALID IN RECFORMAT
C                                                ------------------
      CHARACTER
     .RECTAG*12,     !TAGCODE                          1 2
     .RECTYPE*1,     !RECOVERY TYPE                    1 2
     .RECGRADE*1,    !GRADE (S-M-L)                    1 2
     .RECSPLOC*7,    !RECOV LOCN - MARINE SPORT ONLY   1
     .RECAGETYP*1,   !SAMPLE AGE    M,A,J,I (EG ADULT)   2
     .RECSITEC*1,    !SITE CODE (H/R)                    2
     .RECSTAT*3,     !STAT AREA                        1
     .RECSLA*4,      !SAMPLE LOCN (ALPHA) **           1
     .RECPS*2,       !PROVINCE/STATE  (RECOV AREA) **  1 2
     .RECCU*1,       !'C'=CDN 'U'=USA (RECOV AREA)     1 2
     .RECVTP*1       !VES-TRUCK-PACKER                 1 2
C
      COMMON /CWTRECCHARBLK/
     .RECTAG,RECTYPE,RECGRADE,RECSPLOC,RECSTAT,RECSITEC,
     .RECAGETYP,RECSLA,RECPS,RECCU,RECVTP
C
      BYTE
     .RECBY,         !BROOD YEAR                       1 2
     .RECREPL,       !REPLICATE CODE                   1 2
     .RECSEX,        !SEX-MAT                          1 2
     .RECOLOUR,      !COLOUR 1-R 2-W 3-MIX             1 2
     .RECWK,         !CATCH FISHING PERIOD (MMW)       1 2*
     .RECLENC,       !LENGTH TYPE CODE 1-FK 2-HYP 3-TOT  1 2
     .RECLENT,       !LENGTH SOURCE 1-ACTUAL 2-MEAN    1 2
     .RECWTC,        !WEIGHT TYPE CODE 1-RND 2-DRESS   1 2
     .RECWTT,        !WEIGHT SOURCE 1-ACTUAL 2-MEAN    1 2
```

```
      .RECRUN,          !RECOVERY RUN TYPE (0-4) +                    2
      .RECGEAR          !GEAR CODE                                  1 2
C
       COMMON /CWTRECBYTBLK/
      .RECBY,RECSEX,RECOLOUR,RECWK,RECLENC,RECWTC,
      .RECRUN,RECGEAR,RECWTT,RECLENT,RECREPL
C
       INTEGER*2
      .RECREGION,       !CATCH REGION                               1 2
      .RECWT,           !WT  99V9  KG                               1 2
      .RECLEN,          !LEN  MM                                    1 2
      .RECSUBAREA,      !GUESS (COMMERCIAL)                         1
      .RECSPECIES       !SPECIES (HART NUM)                         1 2
C
       COMMON /CWTRECI2BLK/
      .RECREGION,RECWT,RECLEN,RECSUBAREA,RECSPECIES
C
       INTEGER*4
      .RECNREC,         !NUMBER OF RECOVERIES                       1 2
      .RECSITE,         !SITE ADDRESS                                 2
      .RECCSPER,        !PERIOD POINTER CATSAM,SPTFAC               1
      .RECCDNSP,        !SPECIES POINTER CATSAM SPTFAC              1 2
      .RECFORMAT,       !RECORD FORMAT TYPE                         1 2
      .RECCHAIN,        !PREVIOUS ADDR - SAME TAG                   1 2
      .RECSCALE,        !SCALE AGE                                  1
      .RECSLN,          !SAMPLE LOCN (NUMERIC) (CDN ONLY)           1
      .RECVTPN          !NUMERIC V-T-P CODE                         1
C
       COMMON /CWTRECI4BLK/
      .RECNREC,RECSITE,RECCSPER,RECCDNSP,RECFORMAT,
      .RECCHAIN,RECSCALE,RECSLN,RECVTPN
C
C     +- PERIOD SAMPLE WAS TAKEN
C     *- WHERE APPLICABLE. EG. FW SPORT ("0" IF NOT)
C    **- NOT CURRENTLY IMPLEMENTED
C------------ END INCLUDED RECOVERY RECORD COMMON ------
```

SOURCE CODE MODULES

Some routines access data via passed arguments. Descriptions for these arguments are available in user source code modules.

Commercial Sample Source Code

NOTE: CSREC is not in a common block as it is only a small matrix. The user could code his own description of it, but it is easier to include the MRP version contained in CSRECCOM.FOR.

SSA:[CWTSYS.CATSAM]CSRECCOM.FOR

```
C-----------------------------------------
C      DESCRIPTION A CATCH-SAMPLE RECORD
C
       INTEGER*4 CSREC(40,10,2)
C       40 PERIODS  10 SPECIES   SAMPLE,CATCH (PIECES)
C
C       SPECIES POINTERS:
C       SP: 1 - 118 SOX   2 - 124 CHINOOK  3 - 115 COHO
C           4 - 112 DOG   5 - 108 PINK     6 - 128 STLHEAD
C           7 - 126 MASU  8 -    N/U       9 - 127 CUTTHROAT
C         10  -     N/U
C
C         END  DESCRIPTION A CSREC
C         ---------------------------
```

Escapement Sample Source Code

SSA:[CWTSYS.ESCAPE]ESRECCOM.FOR

```
C-----------------------------------------
C      DESCRIPTION AND DECLARATION OF AN ESCAPEMENT
C      "CATCH/SAMPLE" RECORD  (VIA CWTESGET)
C
      INTEGER*4 ESREC(2,10,4,6)
C
C     RANDOM/SELECT, SPECIES,   SAM-AGE,   1-SAMPLE
C       1      2                          -TOT RUN(RANDOMS)
C                                          3-SUM NO-PINS
C                                          4-SUM LOST-PINS
C                                            (IE LP+CO+BI)
C                                          5-SUM NO-DATAS
C                                          6-SUM "KNOWN" TAGS
C       SAM-AGE POINTERS:
C       SEE SUBROUTINE CWTAJIM
```

```
C
C       SPECIES POINTERS:
C       SP: 1 - 118 SOX   2 - 124 CHINOOK  3 - 115 COHO
C           4 - 112 DOG   5 - 108 PINK     6 - 128 STLHEAD
C           7 - 126 MASU  8 - 000 UNKNOWN  9 - 127 CUTTHROAT
C          10  -     N/U
C
C         END  DESCRIPTION A ESREC
C         ----------------------------
```

Random Recovery Common Block

SSA:[CWTSYS.RECOVERED]RECSUMCOM.FOR

```
C-----------------------------------------
C      DESCRIPTION OF RECOVERED TAGS MATRIX
C     *** NOTE USE OF SPECIES "POT" 8. FOR UNKNOWN SPECIES
C     *** THIS IS DIFFERENT THAN THE OTHER FORMS OF THIS
C         MATRIX.
C     *** DOES NOT INCLUDE "FRESHWATER" CATCH
C         REGIONS.(CR.GE.90)
C     *** INCLUDES ONLY RANDOM RECOVERIES.
C         RECOVERY TYPES 0 THRU 9, BLANK, L, D, Z, R
C
       INTEGER*4 SUMREC(54,10,4)
C      GIVEN A CR AND YR     54 PERIODS  10 SPECIES
C
C      1-NP(NO-PIN) 2-LP(LOST-PIN (INCL CO'S (COLOUR) &
C      BI'S(BINARY))
C      3-ND (NO-DATA)  4-KNOWNS
C
C         NOTE: "RE-" (UNREAD RARE EARTH) NOT
C         INCLUDED.ANYWHERE.
C
C       SPECIES POINTERS:
C       SP: 1 - 118 SOX   2 - 124 CHINOOK  3 - 115 COHO
C           4 - 112 DOG   5 - 108 PINK     6 - 128 STLHEAD
C           7 - 126 MASU  8 - 000 UNKNOWN*
C           9 - 127 CUTTHROAT
C          10  -     N/U
C
C         END  DESCRIPTION A NON-SUM
C         ----------------------------
```

Sport Recovery Matrix

SSA:[CWTSYS.CATSAM]SPTFACREC.FOR

```
C-------------------------------------------------
C
```

```
C     DESCRIPTION OF THE SPORT FACTOR MATRIX
C
C     VIA CWTMI2 RTN...
C
      REAL*4 SPTFAC(25:29,12,10,2)
C
C                  CR   MM SP      1 - ESTIMATED MARKS
C                                  2 - OBSERVED
C                                      RECOVERIES *
C      A NEGATIVE (-1.0) IN ESTIMATED MARKS INDICATES
C      SYSTEM HAS NO SAMPLE DATA FOR THIS CR,MM,SP.
C      A ZERO INDICATES A SAMPLE WITH 0 MI
C
C      * - OBSERVED RECOVERIES EXCLUDES NO-PINS
C
C       SPECIES POINTERS:
C
C       SP: 1 - 118 SOX   2 - 124 CHINOOK  3 - 115 COHO
C           4 - 112 DOG   5 - 108 PINK     6 - 128 STLHEAD
C           7 - 126 MASU  8 -    N/U
C           9 - 127 CUTTHROAT
C          10  -     N/U
C
C         END  DESCRIPTION SPORT FACTOR MATRIX
C         ----------------------------
```

Sport Mark Incidence Counts

SSA:[CWTSYS.CATSAM]CMIREC.FOR

```
C-----------------------------------------
C      DESCRIPTION OF MI MATRIX FOR CDN SPORT
C      MARK INCIDENCE COUNTS. (OBSERVED)
C
      INTEGER*4 MI(25:29,12,10)
C
C       25:29   CDN SPORT CATCH REGIONS..
C       12 PERIODS  (MONTHS)   10 SPECIES
C      A NEGATIVE (-1) IN OBSERVED MARKS INDICATES SYSTEM
C      HAS NO SAMPLE DATA FOR THIS CR,MM,SP.
C      A ZERO INDICATES A SAMPLE WITH 0 MI.
C
C                  USES "CDN" NUMBERS.SEE SUBRTN SPECIES.
C
C       SP: 1 - 118 SOX   2 - 124 CHINOOK  3 - 115 COHO
C           4 - 112 DOG   5 - 108 PINK     6 - 128 STLHEAD
C           7 - 126 MASU  8 -    N/U
C           9 - 127 CUTTHROAT
C          10  -     N/U
C
C         END  DESCRIPTION MARK INCI MAT
```

Catch Sample Area Matrix

```
C------------ COMMON BLOCK FOR CATCH/SAMPLE BY STATAREA
C
C                    USED BY CWTSTATCS
C
      INTEGER*4 STATREC(5,10,2)    !GN-SN-TR-FT-TN,SPECIES, 1-SAMP,2-CAT
C                                    !TN=TOT NET
C                                    !FT NOT SEASONALLY ADJUSTED
C
      INTEGER*4 HARTPNT(108:128)   !SPECIES PNTRS BY HART.
C                                    !FILLED ON FIRST CALL
C                HARTPNT(HART)=STATREC SPECIES INDEX
C
      INTEGER*4 GRPNT(10:33)       !GR INDEXS, WHERE VALID
C                                  !FILLED ON FIRST CALL
C              GRPNT(MRPGR)=STATREC GR INDEX
C
      INTEGER*4 STATGR(5)          !BOATS BY GR
      INTEGER*4 STATDF(5)          !NET GR = DELIVERIES
C                                  !TROLL GR = BOAT-DAYS
      INTEGER*4 STATDO(3)          !DAYS OPEN (Grs 1,2,3)
C
      COMMON /STATCATSAMIBLK/
     .STATREC,HARTPNT,GRPNT,STATGR,STATDF,STATDO
C
C---------------------- END STAT CATCH/SAMPLE BLOCK
```

SUBROUTINE TABLES

MRP tables are formatted in two ways. The table is either 1) carried internally within the subroutine i.e. gear codes are carried within the subroutine CWTGRNAM.FOR or, 2) the tables are carried within a library where the program accesses them i.e. the program CWTPAREA.FOR calls the production area table carried in the library under the name PRODAREA.TAB.

The tables below are carried internally within the subroutines. Library routines for the remaining tables will follow in the section on Library Tables.

Table Type Chart

TABLE TYPES	SUBROUTINE NAME	TABLE CODES
Escapement age codes	CwtAJim	AGE CODES A - adult J - jack I - immature M - mature
Gear type codes	CwtGrNam	GEAR CODES 10 - GILLNET 15 - MIXED NET 20 - SEINE NET 30 - GENERAL TROLL 31 - FREEZER TROLL 32 - DAY TROLL 33 - ICE TROLL 07 - SPORT
Rearing type	CwtRearTyp	REARING CODES R - Wild Fed H - Hatchery W - Wild U - Unknown
Colour code	CwtColour	COLOUR CODES NUM. COLOUR 72 BL - BLUE 73 BR - BROWN 93 CY - CHROME YELLOW 99 DB - DARK BLUE

		95 GD - GOLD
		97 GM - METALLIC GREY
		75 GN - GREEN
		74 GY - GREY
		96 LA - LAVENDER
		76 LB - LIGHT BLUE
		77 LG - LIGHT GREEN
		94 MG - MEDIUM GREEN
		87 MX - MIXED
		78 OR - ORANGE
		82 PK - PINK
		83 PU - PURPLE
		84 RD - RED
		98 VI - VIOLET
		70 WH - WHITE
		80 XR - OXIDE RED
		81 XY - OXIDE YELLOW
		86 YW - YELLOW
		## ## - SOLIDS
		$$ $$ - STRIPES
Nontagcodes	CwtNonTag	NONTAGCODES CODES
		NO-PIN or NO PIN
		R-EARTH or R EARTH
		LOST-PIN or LOST PIN
		NO-DATA or NO DATA
		COLOUR
		BINARY
Release Stage	CwtStage	STAGE CODE
		EE Eyed Egg
		UF Unfed Fry
		FF Fed Fry
		FFS Fed Fry Seapen
		UFC UF Fry Channels
		FFL Fed Fry Lake
		YE Yearlings
		SM Smolts
		SSM Smolts Seapen
Sexual Maturity	CwtSexMat	MATURITY CODES
		1 = Immature Female
		2 = Mature Female
		3 = Immature Male
		4 = Mature Male
		5 = Unknown

Grade	CwtSml	GRADE CODES
		S - Small
		M - Medium
		L - Large
		U - Unknown
Stock Type	CwtStkTyp	STOCK CODES
		H - Hatchery
		W - Wild
		M - Mixed
		U - Unknown
Agency microcode	MicroTran	TABLE

This table carries all the agency codes and the country they are affiliated with. Please refer to **Appendix A** for further references.

1st 2 characters = agency code
next 4 characters = agency
next 3 characters = country
last 2 characters = coordinator code

01WDF USA04
02CDFOCDN03
03NMFSUSA
05FWS USA07
06CDFGUSA08
07ODFWUSA05
08CDFRCDN03
09ODFWUSA05

Run type name	RunType	RUN TYPES
		1 = SPRING
		2 = SUMMER
		3 = FALL
		4 = WINTER
		5 = HYBRID
		6 = LANDLOCKED
Sample location	SamLoc	SAMPLE LOCATIONS
		01 = STEVESTON
		02 = VANCOUVER

03 = VICTORIA
04 = NAMU
05 = PRINCE RUPERT
06 = PORT HARDY
07 = TOFINO
08 = UCLUELET
09 = HISTORICAL
10 = HISTORICAL
11 = HISTORICAL
12 = GEORGIA STRAIT
14 = WINTER HARBOUR
15 = MASSETT
16 = CAMPBELL RIVER
17 = COURTENAY/COMOX
18 = NANAIMO/FRENCH CR.
19 = WHITEHORSE
20 = DAWSON CREEK
21 = TEST FISHERY
22 = EXPORT
23 = BAMFIELD

Species code	Species	HART SPECIES CODES
		108 = PINK
		112 = CHUM
		115 = COHO
		118 = SOCKEYE
		124 = CHINOOK
		126 = MASU
		127 = CUTTHROAT
		128 = STEELHEAD
Study type	StudyType	STUDY TYPES
		W = WILD
		P = PRODUCTION
		M = MONITORING
		B = BOTH PRODUCTION & MONITORING
		E = EXPERIMENTAL
		U = UNKNOWN

LIBRARY TABLES

These tables are loaded from the programs library on the first call (UNIT 100), and these are documented in detail below. They include the production area table, the catch region table, the sport catch region table and the program type table.

Catch Region Name Table

Table Source File Name : REGNAM.TAB
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Purpose of Table :
This table is used for looking up catch region abbreviated names given the catch region number.

Called by:
[CWTSYS.TABLES]CWTREGNAM.FOR

Table Element	Format	Element Description
FLAG(200)	Byte	Set to 1 if catch region I is read in.
NAMES(200)	A20	The abbreviated catch region names.

Element Relationships or Further Description

CODE	CATCH REGION NAME	(STAT AREAS)
0001	NWTR	NW Vancouver Is. Troll 25 - 27)
0002	SWTR	SW Vancouver Is. Troll (21, 23, 24)
0003	WOT	Washington Oregon Troll (historical)
0004	GSTR	Georgia Strait Troll (13 - 18, 29)
0005	CTR	Central Troll (historical)
0006	NTR	Northern Troll (1 - 5)
0007	ATR	Alaska Troll (historical)
0008	FGN	Fraser Gillnet (29)
0009	NN	Northern Net (1 - 5)
0010	GSN	Georgia Strait Net (14 - 18)
0011	JSN	Johnstone Strait Net (11, 12)
0012	CN	Central Net
0013	JFN	Juan de Fuca Net (20)
0014	JFTR	Juan de Fuca Troll
0015	NWTR & CTR	NW Vancouver Is. & Central Troll
0017	NWTR & SWTR	NW & SW Vancouver Is. Troll
0018	NTR &CTR	Northern and Central Troll
0019	JSN & CN	Johnstone Strait and Central Net
0020	NWVN	NW Vancouver Is. Net (25, 26, 27)
0021	SWVN	SW Vancouver Is. Net (21 - 24)
0025	NSPT	Northern Sport (1 - 5)
0026	CSPT	Central Sport (6 - 12, 30)
0027	WSPT	Westcoast Sport (21, 23 - 27)
0028	GSPT	Georgia Strait Sport (13-20, 28, 29)
0029	OFW	Freshwater Sport
0033	NN & CN	Northern & Central Net
0034	GSTR & CTR	Georgia Strait and Central Troll
0036	YKN	Yukon Net
0037	JFN & GSN	Juan de Fuca & Georgia St. Net
0045	JSN & GSN	Johnstone St. & Georgia St. Net
0046	FGN & GSN	Fraser Gillnet & Georgia St. Net

0047	AN	Alaska Net (historical)
0048	BC	British Columbia
0053	GSTR & SWTR	Georgia St. & SW Vancouver Is. Troll
0056	NCTR	North Central Troll (6 - 9, 30)
0057	SCTR	South Central Troll (10 - 12)

Program Type Table

Source File Name : PROGTYP.TAB
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Purpose of Table :
This table is used for looking up program names given the program code.

Called by:
[CWTSYS.TABLES]CWTPROGTYP.FOR

Format	Element Description
A100	1st 4 characters are the production area code, the remaining fields are the production area name

Element Relationships or Furthur Description

CODE	DESCRIPTION
CDP	Community Economic Development
PIP	Public Involvement Program
RES	Research
OPS	Enhancement Operations
SPU	Small Projects Unit
PRV	Province of BC

Production Area Table

Source File Name : PRODAREA.TAB
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Purpose of Table :
This table is used for looking up prodarea names given the prodarea code.

Called by:
[CWTSYS.TABLES]CWTPAREA.FOR

Format	Element Description
A10	1st 4 characters are the production area code, the remaining fields are the production area name

Element Relationships or Furthur Description

CODE	DESCRIPTION
NASS-	NASS RIVER - NASS RIVER & TRIBS (STAT 3)
SKNA-	SKEENA RIVER - SKEENA RIVER & TRIBS (STAT 4)
QCI -	QUEEN CHARLOTTE ISLANDS - STAT AREA 2
NCST-	NORTH COAST - STATS 3 - 5 EXCEPT SKNA & NASS
CCST-	CENTRAL COAST - STATS 6 - 8
RIVR-	RIVERS/SMITH INLETS - STATS 9 - 10
NWVI-	NORTHWEST VANCOUVER ISLAND - STATS 25 - 27
SWVI-	SOUTHWEST VANCOUVER ISLAND - STATS 20 - 24
JNST-	JOHNSTONE STRAIT - STATS 11 - 13
GSVI-	GEORGIA STRAIT VANCOUVER ISLAND - STATS 14,17,18, 19
GSML-	GEORGIA STRAIT MAINLAND - STATS 15, 16, 28, 29
LWFR-	LOWER FRASER RIVER - FRASER RIVER & TRIBS BELOW HOPE
THOM-	THOMPSON RIVER - THOMPSON RIVER & TRIBS
UPFR-	UPPER FRASER RIVER - FRASER RIVER & TRIBS ABOVE HOPE
KOOT-	KOOTENAYS
YUKN-	YUKON
WA01-	WASH. MANAGEMENT AREA 1
WA02-	WASH. MANAGEMENT AREA 2
WA03-	WASH. MANAGEMENT AREA 3
WA04-	WASH. MANAGEMENT AREA 4
WA05-	WASH. MANAGEMENT AREA 5
WA06-	WASH. MANAGEMENT AREA 6
UPWA-	UPPER WASHINGTON
LWWA-	LOWER WASHINGTON
SNAK-	SNAKE RIVER
HEAD-	HEAD WATERS (COLUMBIA)
BRGT-	BRIGHTS (COLUMBIA)
LOCO-	LOWER COLUMBIA
WILL-	WILLIAMETTE
DESC-	DESCHUTES
UPOR-	UPPER OREGON
LWOR-	LOWER OREGON
WEAK-	WESTERN ALASKA
NOAK-	NORTHERN OUTSIDE ALASKA
NIAK-	NORTHERN INSIDE ALASKA
SOAK-	SOUTHERN OUTSIDE ALASKA
SIAK-	SOUTHERN INSIDE ALASKA
CALI-	CALIFORNIA
SACR-	SACRAMENTO
UNKN-	TERRA INCOGNITA
UNAK -	UNKNOWN ALASKA
UNWA -	UNKNOWN WASHINGTON

UNOR - UNKNOWN OREGON
UNCA - UNKNOWN CALIFORNIA
UNBC - UNKNOWN BRITISH COLOMBIA

Sport Catch Region Table

Table Source File Name : SPSTATCR.TAB
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Purpose of Table :
This table is used to find out catch region for statistical area and check that the catch region is a sport region.

Table Element	Format	Element Description
STAB(100)	A3	Array of statistical area codes.
CTAB(100)	I4	Array of corresponding catch region numbers.

Element Relationships or Furthur Description

Statistical Area	Sport Catch Region
STAT AREA= 001	CR=25
STAT AREA= 003	CR=25
STAT AREA= 004	CR=25
STAT AREA= 005	CR=25
STAT AREA= 006	CR=26
STAT AREA= 007	CR=26
STAT AREA= 008	CR=26
STAT AREA= 009	CR=26
STAT AREA= 010	CR=26
STAT AREA= 011	CR=26
STAT AREA= 012	CR=26
STAT AREA= 013	CR=28
STAT AREA= 014	CR=28
STAT AREA= 015	CR=28
STAT AREA= 016	CR=28
STAT AREA= 017	CR=28
STAT AREA= 018	CR=28
STAT AREA= 019	CR=28
STAT AREA= 020	CR=28
STAT AREA= 021	CR=27
STAT AREA= 023	CR=27
STAT AREA= 024	CR=27
STAT AREA= 025	CR=27
STAT AREA= 026	CR=27
STAT AREA= 027	CR=27
STAT AREA= 028	CR=28
STAT AREA= 029	CR=28

```
STAT AREA= 02E    CR=25
STAT AREA= 02W    CR=25
STAT AREA= 030    CR=26
STAT AREA= 0FW    CR=29
STAT AREA= 29B    CR=28
STAT AREA= 29C    CR=28
STAT AREA= 29D    CR=28
STAT AREA= 29E    CR=28
```

SUBROUTINES

Varied and detailed subroutines have been written in order that programmers/users can access the database without extensive knowledge of the systems files and structure. Below is a short explanation of the subroutines used. Following that is the subroutine hierarchy where all subroutines are listed and those subroutines calling other routines are shown. This list is followed by those subroutines using limited unit numbers reserved for the system. User programs must not use units 60 through 100 as they are reserved for current and future system usage. A detailed description of each subroutine, its purpose, arguments used etc. is then documented.

Subroutine Descriptions

CWTAJIM	Handles catch/sample age pointers for escapement.
CWTCALWEK	Calculates sport recovery date.
CWTCHAIN	Performs site chaining through the release master file.
CWTCHAINS	Given site # - returns vector of starting record numbers for release chaining.
CWTCMSTCR	Given statarea code and gear code; returns the catch region.
CWTCOLOUR	Converts numeric tagcode to alpha colour tagcode and vice versa.
CWTCRWK	Given recovery year, catch region, and fishing period; returns the start record and number of records to read in the recovered tag codes file.
CWTCSGET	Given the recovery year and catch region; returns the catch sample record (40 fishing periods, 10 species and catch/sample number.
CWTD3D4	Pass D3D4 of tagcode and get sequence number.
CWTESGET	Given site, runtype and year; returns escapement catch/sample record.
CWTGRNAM	Given gear code returns gear name.
CWTHISITE	Returns current valid highest site number.
CWTLISTREL	Formats release record for output.
CWTMI1	Given year; returns Canadian mark incidence counts for all sport catch regions (25:29).
CWTMI2	Given recovery year; returns estimated marks and observed recoveries for the sport catch.
CWTMMW	Converts dates to fishing periods.
CWTMMW1	As above but covers different opening periods.
CWTNONGET	Given the recovery year and tagcode will return the starting record number in the recovered tag codes file.

CWTNONOPN	Used by CWTNONGET to open address files.
CWTNONTAG	Verifies valid non-tag codes and returns species code.
CWTOPCHN	Opens sites chain index file for accessing the finclip or CWT release master file by site.
CWTOPMST	Opens release master and returns number of records in file.
CWTPAREA	Given prodarea code, returns number of valid codes on file and the prodarea name.
CWTPROGTYP	Given program code (not SEP) will return program name and number of table entries.
CWTREARTYP	Given rearing type code returns rearing name and number of table entries.
CWTRECRD	Reads recovery records.
CWTREGNAM	Given catch region number, returns catch region name and vice versa.
CWTRELEASE	Reads the release file.
CWTRELGET	Reads the release master data file (tagcode access).
CWTRELRD	Reads from release master and stores it in the common block (sequential access).
CWTSEXMAT	Determines sex maturity code.
CWTSITGET	Given the site number and option code; returns the site name, province/state code and prodarea code.
CWTSML	Given the weight, dress code, and gear code; returns fish grade for chinooks.
CWTSPORTLOC	Pass sport location - returns information on area and name.
CWTSPSTCR	Given statarea code; returns sport catch region.
CWTSTAGE	Given the release stage code: returns the release stage name.
CWTSTATCS	Pass statarea, year, period and common block data is returned.
CWTSTKTYP	Converts stock type codes.
CWTSUM1	Adjusts for NO-DATA's and LOST-PINS.
CWTVTP	Returns sample source information.
FINCHAINS	Pass site # - returns vector of starting record number for release chaining.
FINOPMST	Opens finclip release master and returns number of records.
FINRELGET	Reads the finclip release master file (release ID access).
FINRELRD	Reads RECNUM from release master and stores it in the common block (sequential access).
FINTOCWT	Moves variables from fin release block into corresponding variables in CWT release block.
MICROTRAN	Given the releasing agency micro-code; returns the agency name code, coordinator code and country code. Given agency, returns country.
REGIONGR	Validates a catch region/gear code combination.
RELSTAGE	Given release stage code; returns name and vice versa.
RUNTYPE	Given the run number; returns the run name and vice versa.
SAMLOC	Verifies sample location numbers.
SPECIES	Given the Hart species code; returns the name and vice versa.
STATSUB	Verifies catch region, statarea, subarea combinations.
STUDYTYPE	Verifies the given study type code and returns the study type name.

Subroutine Hierarchy

The subroutines are listed below with indented programs to show which other subroutines are being called within the main subroutine.

```
  CwtAJim
  CwtCalWek
  CwtChain
     CwtOpMst
     CwtOpChn
     CwtRelReader
* CwtChains
  CwtCmStCr
     RegionGr
  CwtColour
  CwtCrWk
  CwtCsGet
* CwtD3D4
  CwtEsGet
  CwtGrNam
  CwtHiSite
* CwtListRel
  CwtMi1
  CwtMi2
  CwtMmw
* CwtMmw1
  CwtNonGet
     CwtNonOpn
  CwtNonOpn
  CwtNonTag
  CwtOpChn
  CwtOpMst
  CwtParea
  CwtProgTyp
  CwtRearTyp
  CwtRecRd
     CwtSpec1
     CwtNonTag
  CwtRegNam
* CwtRelease
  CwtRelGet
     CwtOpMst
     CwtRelReader
  CwtRelRd (must call Cwtopmst first)
* CwtSexMat
  CwtSitGet
  CwtSml
* CwtSportLoc
  CwtSpStCr
  CwtStage
* CwtStatCs
  CwtStkTyp
```

```
  CwtSum1
* CwtVtp
* FinChains
  FinOpMst
  FinRelGet
     FinOpMst
     FinRelReader
  FinRelRd (must be opened by Finopmst first)
* FinToCwt
  MicroTran
  RegionGr
  Relstage
  RunType
  SamLoc
  Species
  StatSub
  StudyType

*  new subroutines
```

Subroutines and Restricted Numbers

SUBROUTINE	UNIT NOS. USED
CWTCHAIN	90, 99
CWTCMSTCR	100
CWTCRWK	94
CWTCSGET	98
CWTESGET	83
CWTMI1	100
CWTMI2	100
CWTNONGET	81
CWTNONOPN	81
CWTOPCHN	99
CWTOPMST	90
CWTPAREA	100
CWTRECRD	97, 92, 93
CWTREGNAM	100
CWTRELGET	90, 95
CWTRELRD	90
CWTSITGET	96, 97
CWTSPSTCR	100
FINOPMST	70
FINRELGET	70, 75
FINRELRD	70
STATSUB	91

CWTAJIM

Source File Name : CWTAJIM.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Handles catch/sample age pointers for escapement recoveries. If character (CAJIM) is passed, number(IAJIM) and name (CNAME) are returned. If number(IAJIM) is passed, character (CAJIM) and name (CNAME) are returned. See ESRECCOM.FOR valid codes.

Arguments Passed :

CAJIM	Character*1	----	Alpha character for escapement age.
IAJIM	Integer*4	----	Numeric escapement age.
CNAME	Character*10	----	The escapement age.
OPT	Integer*4	----	The option code.

Arguments Returned :

CAJIM	Character*1	----	Alpha character for escapement age.
IAJIM	Integer*4	----	The numeric escapement age.
CNAME	Character*10	----	The escapement age.
ERROR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtaJim( CAJIM,IAJIM,CNAME,OPT,ERROR )
```

Option Codes / Description :

OPT = 1	----	Given alpha age code, returns numeric age and description.
OPT = 2	----	Given numeric age code, returns alpha code and description.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Errors

CWTCALWEK

Source File Name : CWTCALWEK.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine calculates the sport recovery date. If day (D), month (M), or year (Y) is passed, week (W) is returned. If month (M), year (Y) and week (W) are passed, day (D) is returned.

Arguments Passed :

D	----	The day.
M	----	The month.
Y	----	The year.
W	----	The calender week.
OPT	----	The option code.

Arguments Returned :

D	----	The day.
M	----	The month.
Y	----	The year.
W	----	The calender week.
ERROR	----	The error code.

Subroutine Call Example :

```
Call Cwtcalwek( D,M,Y,W,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Given day, month and year; calender week is returned.
OPT = 2	----	Given month, year and week, last day of week is returned.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Error in date.
ERR = 2	----	Option error.

Special Considerations :

1. Sport data are only expandable at the month level, though sport recoveries may show an actual recovery week.
2. Calender weeks are not the same as statistical weeks (statweeks). They do not cross month boundaries and may be less than a 7 day period. Week 1 ends the first Saturday of the month. Last week ends the last day of the month.

CWTCHAIN

Source File Name : CWTCHAIN.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine retrieves records from the release master file (ADDR) by hatchery (OPT=1), release site (OPT=2) and stock (OPT=3) and puts it into the release master common blocks. Records from the CWT release file (JOPT=0) or finclip release file (JOPT=1) can be retrieved in reverse tagcode sequence.

Arguments Passed :

ADDR	Integer*4	----	Record number in the sites file used to find the record in the release master file.
OPT	Integer*4	----	Option code.
JOPT	Integer*4	----	Option code.

Arguments Returned :

ADDR	Integer*4	----	Record number in the release file.
ERR	Integer*4	----	Error result code.

Subroutine Call Example :

```
Call CwtChain( ADDR,OPT,JOPT,ERR )
```

Option Codes / Description :

```
OPT = 1  ---- Return last record same hatchery.
OPT = 2  ---- Return last record same release site.
OPT = 3  ---- Return last record same stock.
JOPT = 0 ---- From CWT release master file.
JOPT = 1 ---- From FINCLIP release master file.
```

Error Condition Codes / Description :

```
ERR = 0  ---- Okay.
ERR = 1  ---- Invalid address ( ADDR must be between 1
              and 9999).
ERR = 2  ---- 'END OF FILE', no record read.
ERR = 4  ---- Chain break: change has been made to the
              release master file and the maintenance
              program has not been run to restructure
              the chains. Record in common block may
              not have been updated. Contact MRP technician.
```

Included Common Blocks :

Source File Name	: RELRECCOM.FOR
Disk:[Directory]	: CWT:[CWTSYS.RELEASE]

Other Subroutines Called :

Source File Name	: CWTOPMST.FOR(NDUM)
Source File Name	: CWTOPCHN.FOR (start pointer)
Source File Name	: CWTRELRD.FOR (RECNUM)
Disk:[Directory]	: CWT:[CWTSYS.TABLES]

Special Considerations :

1. This routine uses Fortran file units 90, 99 (internal routines only).
2. A change in option (OPT) or address (ADDR), does a "reset" in reverse tagcode sequence.
3. JOPT is applied only on 1st call.

CWTCHAINS

Source File Name : CWTCHAINS.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine will return the vector of starting record numbers for release record chaining (HRSNUMS) when passed the hatchery, release or site number (SITE).

Arguments Passed :

SITE	Integer*4	----	Hatchery, release or stock site.
OPT	Integer*4	----	The option code.

Arguments Returned :

HRSNUMS(3)	Integer*4	----	Starting record number for hatchery, release and stock for this site code.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtChains(SITE,HRSNUMS,OPT,ERR)
```

Option Codes/Descriptions:

OPT = 1	----	Pass site number; return vector of starting record numbers to begin chaining hatchery, release or stock site.

Error Conditions:

ERR = 0	----	The site number was okay.
ERR = 2	----	Invalid site number.

Special Considerations:

1. This program uses unit 99.

CWTCMSTCR

Source File Name : CWTCMSTCR.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
For commercial recoveries; statistical area (STAT) and gear (GR) are passed and catch region (CR) is returned.

Arguments Passed :

STAT	Character*3	----	The statistical area (statarea) code.
GR	Integer*4	----	The numeric gear code.
OPT	Integer*4	----	The option code.

Arguments Returned :

CR	Integer*4	----	Catch region returned.
ERR	Integer*4	----	The error code.

Subroutine Call example :

```
Call CwtCmStCr( CR,STAT,GR,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Statarea and gear are passed; catch region is returned.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Unknown statarea.
ERR = 2	----	Gear code invalid.
ERR = 3	----	Catch region for this statarea/gear code does not exist.
ERR = 99	----	A given statarea/gear code combination points to more than one catch region. Writes error log to FOR100.DAT. Call SSA support group.

Other Subroutines Called :
Source File Name : REGIONGR.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Other Inputs :

Source File Name : STATSUB.DAT
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Other Outputs :
File Name : FOR100.DAT (see error code 99).

Special Considerations :
1. (GR) argument must contain new gear codes.
2. Uses Fortran file unit 100 and leaves closed.
3. On the first call, the program builds look up tables from the STATSUB.DAT file using the subroutine REGIONGR and uses the tables to test for 'non-unique' statarea/gear code combinations (see error code 99).
4. STATSUB.DAT is used for verifying data entry records.

CWTCOLOUR

Source File Name : CWTCOLOUR.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine or Program :
Converts numeric colour tags (NTAG) to alpha colour tags (ATAG) and vice versa.

Arguments Passed :

NTAG	Character*8	----	The numeric colour tagcode.
ATAG	Character*8	----	The alpha colour tagcode.
OPT	Integer*4	----	The option code.

Arguments Returned :

NTAG	Character*8	----	The numeric colour tagcode.
ATAG	Character*8	----	The alpha colour tagcode.
ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtColour( NTAG,ATAG,OPT,ERR ).
```

Option Codes / Description :

OPT = 1	----	Numeric tag passed, alpha tag returned.
OPT = 2	----	Alpha tag passed, numeric tag returned.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Conversion error.

Special Considerations :

1. Translation stops if/when a "*" suffix is found (* indicates repeated uses of the same code).
2. ## (solids) and $$ (stripes) are valid tag prefixes.

CWTCRWK

Source File Name : CWTCRWK.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given a recovery year (YR), catch region (CR), and fishing period (IPR); this routine gives starting address START) and number of records to read (NREC) for tag recovery records in recovery file RECTAGSyy.DAF.

Arguments Passed :

YR	Integer*4	----	The recovery year.
CR	Integer*4	----	The numeric catch region code.
IPR	Integer*4	----	The fishing period.
OPT	Integer*4	----	The option code.

Arguments Returned :

START	Integer*4	----	Starting record number in tagcode file.
NREC	Integer*4	----	The number of records to read.
ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtCrWk( YR,CR,IPR,START,NREC,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Given recovery year, catch region, and fishing period, returns starting address and number of records.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	No data for this recovery year.
ERR = 2	----	Catch region is not valid.
ERR = 3	----	Fishing period is not valid.

Special Considerations :

1. This routine uses Fortran unit number 94 (internal routine only).
2. Sample code : DO I=START, START+NREC-1.

CWTCSGET

Source File Name : CWTCSGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Gets a catch/sample record (CSREC) for catch region (CR) and recovery year (YR) combination.

Arguments Passed :

CR	Integer*4	----	The numeric catch region code.
YR	Integer*4	----	The recovery year .
OPT	Integer*4	----	The option code.

Arguments Returned :

CSREC	Integer*4(40,10,2)	----	The matrix of catch sample ratios.
ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtCsGet( CR,YR,CSREC,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Given recovery year; catch region returns the catch/sample record.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	No data for this recovery year.
ERR = 2	----	No data for this catch region.

Included Source Code :
Source File Name : CSRECCOM.FOR

Special Considerations :
1. This routine uses Fortran unit number 98.

CWTDW3D4

Source File Name : CWTD3D4.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine obtains the sequence number from the data3/data4 portion of a sequential coded-wire tag.

Arguments Passed :

D3	Integer*4	----	Data 3.
D4	Integer*4	----	Data 4.
OPT	Integer*4	----	The option code.

Arguments Returned :

SEQNUM	Integer*4	----	Sequential number.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtD3D4(D3,D4,SEQNUM,OPT,ERR)
```

Option Codes/Descriptions:

OPT = 1	----	Pass sequential D3,D4 to get sequence number.

Error Conditions:

ERR = 0	----	OK
ERR = 1	----	Bad D3/D4 or the option is not equal to 1.

Special Considerations :

1. The sequential number is calculated using an algorithm supplied by Northwest Marine Technology.

CWTESGET

Source File Name : CWTESGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine gets an escapement catch/sample record (ESREC) for site (ISITE), runtype (RUN) and year (YR).

Arguments Passed :

ISITE	Integer*4	----	The numeric site address.
CHR	Character*1	----	Either H (hatchery) or R (relsite).
RUN	Integer*4	----	The runtype (1 - 6).
YR	Integer*4	----	The year of recovery.
OPT	Integer*4	----	The option code.

Arguments Returned :

ESREC	Integer*4(2,10,4,6)	----	The escapement catch sample matrix.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call Cwtesget( ISITE,CHR,RUN,YR,ESREC,OPT,ERR )
```

Option Codes / Description :
OPT = 1 ---- Given site address, hatchery (H) or release site (R), run and year; get the escapement catch sample record.

Error Condition Codes / Description :
ERR = 0 ---- No errors.
ERR = 1 ---- No data for this year.
ERR = 2 ---- No data for this site and runtype.

Included Source Code :
Source File Name : ESRECCOM.FOR
Disk:[Directory] : SSA:[CWTSYS.ESCAPE]

Special Considerations :
1. This routine uses Fortran unit number 83 (internal routine only).

CWTGRNAM

Source File Name : CWTGRNAM.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given either old or new gear codes (GR); gear name (NAME) is returned.

Arguments Passed :

GR	Integer*4	----	Old or new numeric gear code.
OPT	Integer*4	----	The option code.

Arguments Returned :

NAME	Character*15	----	Alpha gear name.
ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtGrNam( GR,NAME,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Old gear code passed; gear name returned.
OPT = 2	----	New gear code passed; gear name returned.
OPT = 3	----	Old gear code passed; new gear code and name returned.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Gear code does not exist.
ERR = 2	----	Option requested is not valid.

CWTHISITE

Source File Name : CWTHISITE.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Will return highest valid site numbers for each of release site, hatchery and stock.

Arguments Passed :

OPT	Integer*4	----	The option code.

Arguments Returned :

HIHAT	Integer*4	----	The highest hatchery code.
HIREL	Integer*4	----	The highest release site code.
HISTK	Integer*4	----	The highest stock code.
ERR	Integer*4	----	The option code.

Subroutine Call Example :

```
Call CwtHiSite( HIHAT,HIREL,HISTK,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Returns current highest valid site numbers.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Report to SSA support group.

Special Considerations :

1. Uses unit 100, leaves it closed (internal routine only).
2. For freshwater recoveries, release site may also be recovery site. The site numbers can then also be used for freshwater recovery, catch and sample data.

CWTLISTREL

Source File Name : CWTLISTREL.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine formats and writes the release record currently in the release common block (RELRECCOM.FOR) to an output file open on logical unit "IO". The routine uses Fortran carriage control. Unit number must be less than 50. This routine saves programmers the task of writing code to put release information into readable form. The data from release is written to the file, as well as labels for each field printed.

Arguments Passed :
IO - output unit

Arguments Returned :

Subroutine Call Example :

```
Call CwtListRel(IO)
```

Included Common Blocks :
Source File Name : RELRECCOM.FOR
Disk:[Directory] : CWT:[CWTSYS.RELEASE]

Other Subroutines Called :
Source File Name : CWTSTAGE.FOR
Source File Name : CWTSITGET.FOR
Source File Name : RUNTYPE.FOR
Source File Name : SPECIES.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

CWTMI1

Source File Name : CWTMI1.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine gets the Canadian mark incidence counts for all Canadian sport catch regions. Pass the year (YR) and the array (MI) is returned.

Arguments Passed :

YR	Integer*4	----	The recovery year.
OPT	Integer*4	----	The option code.

Arguments Returned :

MI	Intger*4(25:29,12,10)	----	Mark incidence matrix (sport catch region, month, species).
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call Cwtmi1( YR,MI,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Pass the recovery year and mark incidence array is returned.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	No recovery data for this year.
ERR = 2	----	Not a valid option.

Included Source Code :
Source File Name : CMIREC.FOR
Disk:[Directory] : CWT:[CWTSYS.CATSAM]

Special Considerations :

1. This routine uses Fortran unit 100 (internal unit only).
2. One call per recovery year will return all data.
3. A -1 in the observed marks indicates system has no sample data for this catch region, month or species.
4. A 0 indicates the sample has no mark incidence.

CWTMI2

Source File Name : CWTMI2.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine gets Canadian estimated marks in the sport catch along with observed recoveries (SPTFAC) when passed the recovery year (YR).

Arguments Passed :

YR	Integer*4	----	The recovery year.
OPT	Integer*4	----	The option code.

Arguments Returned :

SPTFAC	Real*4(25:29,12,10,2)	----	Recovery matrix .
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CWTMI2( YR,SPTFAC,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Pass the recovery year and sport recovery is returned.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	No recovery data for this year.
ERR = 2	----	Not a valid option.

Included Source Code :
Source File Name : SPTFACREC.FOR
Disk:[Directory] : CWT:[CWTSYS.CATSAM]

Special Considerations :
1. This routine uses Fortran unit 100 (internal unit only) and leaves it closed.
2. One call per recovery year will return all data.
3. A -1 in the observed marks indicates system has no sample data for this catch region, month or species.
4. A 0 indicates the sample has no mark incidence.
5. Observed recoveries excludes NO-PINS.

CWTMMW

Source File Name : CWTMMW.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
For Canadian commercial statweeks, converts day (D), month (M), year (Y) to catch month (MM), week (W) and fishing period (PERIOD); or it takes catch month (MM), week (W) and returns fishing period (PERIOD); or converts fishing period (PERIOD), to month (MM) and week (W); or converts year (Y), month (MM) and week (W) to day (D) and month (MM).

Arguments Passed :

D	Integer*4	---- The day of the month.
M	Integer*4	---- The month of the year.
Y	Integer*4	---- The year.
MM	Integer*4	---- The catch month.
W	Integer*4	---- The catch week.
PERIOD	Integer*4	---- The fishing period.
OPT	Integer*4	---- The option code.

Arguments Returned :

MM	Integer*4	---- The catch month.
W	Integer*4	---- The catch week.
PERIOD	Integer*4	---- The fishing period.
ERR	Integer*4	---- The error result code.

Subroutine Call Example :

```
Call CwtMMW( D,M,Y,MM,W,PERIOD,OPT,ERR )
```

Option Codes / Description :

OPT = 1	---- Day, month, year passed; catch month, week, fishing period returned.
OPT = 2	---- Catch month and week passed; fishing period returned.
OPT = 3	---- Fishing period passed; catch month and week returned.
OPT = 4	---- Year, catch month and week passed; day and month returned. (Returns data for the ending period).

Error Condition Codes / Description :

ERR = 0	---- No errors.
ERR = 1	---- Input date error.
ERR = 2	---- OPT = 2,4 catch month and week is not valid.
ERR = 3	---- OPT = 3, fishing period not valid (1-39).

CWTMMW1

Source File Name : CWTMMW.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This is a modified version of CWTMMW. It performs all functions of the earlier routine (Canadian weeks 031 thru 114, periods 1 thru 40), and can also process dates based on a 54 period year, weeks 011 thru 126. Changes were also made to make this routine run faster that the earlier version. This routine is intended to replace CWTMMW although both will continue to be supported.
For Canadian commercial statweeks, converts day (D), month (M), year (Y) to catch month (MM), week (W) and fishing period (PERIOD); or it takes catch month (MM), week (W) and returns fishing period (PERIOD); or converts fishing period (PERIOD), to month (MM) and week (W); or converts year (Y), month (MM) and week (W) to day (D) and month (MM).

Arguments Passed :

D	Integer*4	---- The day of the month.
M	Integer*4	---- The month of the year.
Y	Integer*4	---- The year.
MM	Integer*4	---- The catch month.
W	Integer*4	---- The catch week.
PERIOD	Integer*4	---- The fishing period.
OPT	Integer*4	---- The option code.

Arguments Returned :

MM	Integer*4	---- The catch month.
W	Integer*4	---- The catch week.
PERIOD	Integer*4	---- The fishing period.
ERR	Integer*4	---- The error result code.

Subroutine Call Example :

```
Call CwtMMW( D,M,Y,MM,W,PERIOD,OPT,ERR )
```

Option Codes / Description :

OPT = 1 ----	Day, month, year passed; catch month, week(MMW), fishing period returned.
OPT = 2 ----	Catch month and week passed; fishing period returned.
OPT = 3 ----	Fishing period passed; catch month and week returned.
OPT = 4 ----	Year, catch month and week passed; day and month returned. (Returns data for the ending period).

OPT = 6 ---- Day, Month year sent for weeks ending Sunday. MMW, period returned for periods 1 - 54 and MMW 011 to 126.
OPT = 7 ---- Day, Month year sent for weeks ending Saturday. MMW, period returned for periods 1 - 54 and MMW 011 to 126.
OPT = 21 ---- MMW sent, period returned (1 thru 54).
OPT = 31 ---- Period sent, MMW returned (1 thru 54).
OPT = 41 ---- Y, MMW sent, D,M returned (Date period ends) weeks ending on Saturday for MMW 011 - 126.
OPT = 42 ---- Y, MMW sent, D,M returned (Date period ends) weeks ending on Sunday for MMW 011 - 126.

Error Condition Codes / Description :
ERR = 0 ---- No errors.
ERR = 1 ---- Input date error.
ERR = 2 ---- From OPT = 2,4,21,41,42 catch month and week is not valid.
ERR = 3 ---- From OPT = 3,31 fishing period is not valid (1-39),(1-54).
ERR = 9 ---- Invalid option.

CWTNONGET

Source File Name : CWTNONGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given recovery year (YR) and non-tag code (TAG); returns the starting address (RECNUM) for non-tag codes (No-Pin, Lost-Pin, No-Data etc.) from the recovery files RECTAGSyy.DAF's.

Arguments Passed :

TAG	Character*8	----	Non-tagcode.
YR	Integer*4	----	Recovery year required.
OPT	Integer*4	----	The option code.

Arguments Returned :

RECNUM	Integer*4	----	Beginning record number for non-tag codes.
ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtNonGet( TAG,YR,RECNUM,OPT,ERR )
```

Option Codes / Description :
OPT = 1 ---- Given tagcode and recovery year; return the starting address.

Error Condition Codes / Description :
ERR = 0 ---- No errors.
ERR = 1 ---- Tagcode not found.
ERR = 2 ---- No tag codes for this recovery year.

Other Subroutines Called :
Source File Name : CWTNONOPN.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :
1. On the first call for the year, all addresses are loaded and saved.
2. Handles up to 50 years per run before there is a table overflow.
3. This routine uses Fortran unit number 81.

CWTNONOPN

Source File Name : CWTNONOPN.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given recovery year (YR) this routine will open the non-tagcode starting chain address file for indexing to the RECTAGSyy.DAF file of recovered tag codes.

Arguments Passed :

YR	Integer*4	----	The recovery year required.
OPT	Integer*4	----	The option code.

Arguments Returned :

ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtNonOpn( YR,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Opens the file recovery file.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	No tags available for this recovery year.

Called From :
Source File Name : CWTNONGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :
1. This routine uses Fortran unit number 81.

CWTNONTAG

Source File Name : CWTNONTAG.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
If the passed tagcode (TAG) is a NO-PIN etc., it and the passed species code (SPECIES) are combined to create a new tagcode i.e. NP-115. Another option takes the passed tagcode (TAG) and validates it as a valid non-tagcode (TAG); then species (SPECIES) is returned.

Arguments Passed :

TAG	Character*8	----	Non-tagcode.
SPECIES	Integer*4	----	Species combined with non-tagcode.
OPT	Integer*4	----	The option code.

Arguments Returned :

TAG	Character*8	----	Valid tagcode/species combination.
SPECIES	Integer*4	----	The species code retrieved from the tagcode.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtNonTag( TAG,SPECIES,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Non-tagcode and species are passed, and a valid non-tagcode is returned.
OPT = 2	----	A valid non-tagcode is passed, and species is returned.
OPT = 3	----	Verifies that the passed tagcode is a valid non-tagcode.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Tagcode is not a valid non-tagcode.
ERR = 2	----	Species passed is an invalid species code.

CWTOPCHN

Source File Name : CWTOPCHN.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine opens the sites chaining index file for CWT or Finclip files.

Arguments Passed :
OPT Integer*4 ---- The option code.

Arguments Returned :
NONE

Subroutine Call Example :
```
Call CwtOpChn (OPT)
```

Option Codes / Description :
OPT = 0 ---- Opens the CWT sites chaining index file.

OPT = -1 ---- Opens the Finclip sites chaining index file.

Error Condition Codes / Description :
NONE

Called From :
Source File Name : CWTCHAIN.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :
1. This routine uses Fortran unit number 99 (internal routine only).
2. OPT = -1 was used in some old programs to open finmark release file.
3. See CWTCHAIN.FOR.

CWTOPMST

Source File Name : CWTOPMST.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine opens the release master file and returns the number of records (NREC) in the file .

Arguments Passed :
NONE

Arguments Returned :

NREC	Integer*4	----	The number of release records in the file.

Subroutine Call Example :

```
Call CwtOpMst( NREC )
```

Error Condition Codes / Description :
NONE

Special Considerations :

1. Record 1 has a null tag code and contains the number of releases (NREC).
2. If NREC is passed as -1 then [FINCLIP.RELEASE]RELEASE.DAF is opened (this option is used only in some old programs for compatibility). NOTE : The finmark subroutine FINOPMST should be used.
3. This routine uses Fortran unit number 90.
4. Sample code : DO I=2, NREC.

CWTPAREA

Source File Name : CWTPAREA.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given the production area (prodarea) code (PROD); returns the number of valid prodareas on file (NPROD), the full name of the prodarea (PNAME), the length of the name (PNAMLEN), and the position in the table (NADD); or given the position in the table will return PROD, NPROD, PNAME and PNAMLEN.

Arguments Passed :

PROD	Character*4	----	The alpha prodarea code.
NADD	Integer*4	----	The numeric table position of a prodarea.
OPT	Integer*4	----	The option code.

Arguments Returned :

NPROD	Integer*4	----	Number of prodarea codes in the table.
PROD	Character*4	----	The alpha prodarea code.
PNAME	Character*80	----	The prodarea name.
PNAMLEN	Integer*4	----	The length of the prodarea name.
NADD	Integer*4	----	Numeric position of a prodarea in the table.
ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtParea( NPROD,PROD,PNAME,PNAMLEN,NADD,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Given prodarea code will return prodarea information.
OPT = 2	----	Given position in the table; will return prodarea information.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	The prodarea code passed is unrecognized.
ERR = 2	----	The table passed is outside of the table range.
ERR = 99	----	"Table Overflow" occurred, call SSA support group.

Other Inputs :

Source File Name : PRODAREA.TAB
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :

1. This routine uses Fortran unit number 100 on the first call then saves the table array in an internal file and closes the unit off.

CWTPROGTYP

Source File Name : CWTPROGTYP.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
When program code (PROG) passed, program name(PROGNAME) and number of table entries (NVAR) is returned; or when table entry number passed (NVAR), program code (PROG) and program name (PROGNAME) returned.

Arguments Passed :

PROG	Character*3	----	The program code.
PROGNAME	Character*40	----	The program name.
NVAR	Integer*4	----	The table entry number.
OPT	Integer*4	----	The option code.

Arguments Returned :

PROG	Character*3	----	The program code.
PROGNAME	Character*40	----	The program name.
NVAR	Integer*4	----	The table entry number.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtProgTyp( PROG,PROGNAME,NVAR,OPT,ERR)
```

Option Codes / Description :

OPT = 1	----	Program code passed, program name and number of entries returned.
OPT = 2	----	Nothing passed, # of entries returned.
OPT = 3	----	Entry number passed, program code and name returned.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Program code not in table (OPT 1).
ERR = 1	----	Entry number not in table (OPT 3).
ERR = 2	----	Invalid option code.
ERR = 3	----	Internal error. NOTIFY SSA Support Group.

Other Inputs :
Source File Name : PROGTYP.TAB

Special Considerations :
1. This routine uses Fortran unit number 100 (internal routine only) and leaves it closed.

CWTREARTYP

Source File Name : CWTREARTYP.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
If release rearing type (CODE) is passed, returns rearing name (NAME) and number of table entries (NVAR); or returns number of entries in table (NVAR); or if passed table entry number (NVAR), will return rearing type (CODE) and rearing name (NAME).

Arguments Passed :

CODE	Character*1	----	The rearing code.
NAME	Character*20	----	The rearing name.
NVAR	Integer*4	----	The table entry number.
OPT	Integer*4	----	The option code.

Arguments Returned :

CODE	Character*1	----	The rearing code.
NAME	Character*20	----	The rearing name.
NVAR	Integer*4	----	The table entry number.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtRearTyp( CODE,NAME,NVAR,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Return rearing type name and NVAR given rearing code.
OPT = 2	----	Returns maximum table entries.
OPT = 3	----	Returns rearing code and name given table entry.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Invalid code passed (option 1).
ERR = 2	----	Invalid NVAR passed (option 3).

CWTRECRD

Source File Name : CWTRECRD.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine reads in the record specified by RECNUM from the recovered tagcode data file RECTAGSyy.DAF and places the record in the recovered tagcodes common blocks. The variable RECCHAIN in the common block is updated with the next record. Recovery records are sequenced by descending time period within descending catch region number. If RECCHAIN is 0 then the last record is in common block.

Arguments Passed :

YR	Integer*4	----	The recovery year.
RECNUM	Integer*4	----	The record number in the file.
OPT	Integer*4	----	The option code.

Arguments Returned :

ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtRecRd( YR,RECNUM,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Read the record (RECNUM) from the recovered tagcode file.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	No recovery file for specified recovery year (YR).
ERR = 2	----	Read error - probably invalid RECNUM.
ERR = 3	----	Record found but the release is no longer in the current release master data file. Species number or brood year may be in error. Contact SSA support group.

Included Common Blocks :
Source File Name : RECTAGCOM.FOR
Disk:[Directory] : CWT:[CWTSYS.RECOVERED]

Other Subroutines Called :

Source File Name : TAGLOOK.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Source File Name : CWTNONTAG.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Source File Name : CWTSPEC1.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :

1. If tag is a non-tagcode then brood year is set to 0 in the recovery record common block.
2. This routine uses Fortran unit number 97. TAGLOOK is called by this routine and uses Fortran unit numbers 92 & 93.
3. Returns the sport location data when present.

CWTREGNAM

Source File Name : CWTREGNAM.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine will return the catch region name (NAME) given the region number (REG); or will return the catch region number (REG) given the catch region name (NAME). The highest catch region number (HIVAL) is always returned.

Arguments Passed :

REG	Integer*4	----	Numeric catch region code.
NAME	Character*20	----	Character name of the catch region.
OPT	Integer*4	----	The option code.

Argument Returned :

REG	Integer*4	----	Numeric catch region code.
NAME	Character*20	----	Character name of the catch region.
HIVAL	Integer*4	----	Highest catch region numeric code.
ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call CwtRegNam( REG,NAME,HIVAL,OPT,ERR )
```

Option Codes / Description :

```
OPT = 1  ----  Return catch region name given CR#.
OPT = 2  ----  Return  CR# given catch region name.
```

Error Condition Codes / Description :

```
ERR = 0  ----  No error.
ERR = 1  ----  Catch region is not in the table.
```

Other Inputs :

```
Source File Name : REGNAM.TAB
Disk:[Directory] : CWT:[CWTSYS.TABLES]
```

Special Considerations :

1. This routine uses Fortran unit number 100 on first call then closes it.

CWTRELEASE

Source File Name : CWTRELEASE.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :

This routine reads tagcodes (TAGCODE) from the release file and returns the release and record number (RECNUM), or it reads the record number (RECNUM) and returns the release record(NREC) and tagcode (TAGCODE); or a sequential reading of the file will return the release record, the record number and the tagcode. The fourth option will close the master and do a reset or allows the user to go between options 1-2 and option 3. This option is also needed for passes on option 3.

Arguments Passed :

TAGCODE	Character*12	----	The tagcode.
RECNUM	Integer*4	----	The record number.
NREC	Integer*4	----	The last record in the file.
OPT	Integer*4	----	The option code.

Arguments Returned :

TAGCODE	Character*12	----	The tagcode.
RECNUM	Integer*4	----	The record number.
NREC	Integer*4	----	Files last record number.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtRelease(TAGCODE,RECNUM,NREC,OPT,ERR)
```

Option Codes/Descriptions:

OPT = 1	----	Reads the tagcode; release record and record number returned.
OPT = 2	----	Reads the record number; tagcode and release record returned.
OPT = 3	----	Sequential read of file; the release, record # and tagcode returned.
OPT = 4	----	Closes the release master file in order to do a reset. This allows users to change between options 1-2 and option 3. This option is also needed between passes on the sequential read.

Error Conditions:

ERR = 0 ---- OK.
ERR = 1 ---- Tag not found.
ERR = 2 ---- No such record/EOF on sequential read.

Included Common Blocks :

Source File Name : RELRECCOM.FOR
Disk:[Directory] : CWT:[CWTSYS.RELEASE]

Other Subroutines Called :

Source File Name : CWTRELREADER.FOR

Special Considerations:

1. On the first call, NREC is returned. Releases are stored in records 2 thru record NREC in the master. Record 1 stores the last record number in the file (NREC).
2. Calling program cannot mix options 1 or 2 with 3.
3. Program is free to mix calls using options 1 and 2.
4. Program uses units 90 and 95.

CWTRELGET

Source File Name : CWTRELGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Does a direct read of the release master data file using the passed tagcode (TAGCODE); or a read of the release master, reading the record following the previous record read.

Arguments Passed :

TAGCODE	Character*8	----	The tagcode.
OPT	Integer*4	----	The option code.

Arguments Returned :

ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtRelGet( TAGCODE,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Given tagcode; read corresponding release record and put into release master common blocks.
OPT = 2	----	Read record following previous record read. Does a sequential read of the file. Gets record following last record accessed by routine on any option. (File is in tagcode sequence).

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Tagcode not found.
ERR = 2	----	End of file, no record read (from option 2).

Included Common Blocks :
Source File Name : RELRECCOM.FOR
Disk:[Directory] : CWT:[CWTSYS.RELEASE]

Other Subroutines Called :

Source File Name : CWTOPMST.FOR (NREC)
Source File Name : CWTRELGET.FOR (LOOK2)
Source File Name : CWTRELRD.FOR (IGET)
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :
1. This routine uses Fortran unit numbers 90 & 95 (internal routine only).

CWTRELRD

Source File Name : CWTRELRD.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine reads record (RECNUM) from the release master data file and stores it in the release master common blocks.

Arguments Passed :
RECNUM Integer*4 ---- Record number.

Arguments Returned :
NONE

Subroutine Call Example :
```
Call CwtRelRd( RECNUM )
```

Option Codes / Description :
NONE

Error Condition Codes / Description :
NONE

Included Common Blocks :
Source File Name : RELRECCOM.FOR
Disk:[Directory] : CWT:[CWTSYS.RELEASE]

Other Inputs :
Source File Name : RELEASE.DAF
Disk:[Directory] : CWT:[CWTSYS.RELEASE]

Special Considerations :
1. This routine uses Fortran unit number 90 (internal routine only).
2. File must be opened by the subroutine CWTOPMST.

CWTSEXMAT

Source File Name : CWTSEXMAT.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine returns sex (SEX) and maturity (MAT) if passed code (CODE); or if passed sex (SEX) and maturity (MAT) will return code (CODE).

Arguments Passed :

SEX	Character*1	----	The sex code.
MAT	Character*1	----	The maturity code.
CODE	Integer*4	----	The sex maturity code.
OPT	Integer*4	----	The option code.

Arguments Returned :

SEX	Character*1	----	The sex code.
MAT	Character*1	----	The maturity code.
CODE	Integer*4	----	The sex maturity code.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtSexMat( CODE,SEX,MAT,OPT,ERR )
```

Option Codes / Description :
OPT = 1 ---- Given code; returns sex and maturity.
OPT = 2 ---- Given sex and maturity; returns code.

Error Condition Codes / Description :
ERR = 0 ---- No errors.
ERR = 1 ---- Invalid arguments passed.

CWTSITGET

Source File Name : CWTSITGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
For a given hatchery or stock or release area code (ADDR) depending on the option code (COD), this routine returns names where:
STR1 contains the name depending on option code.
STR2 contains the production area code.
STR3 contains the province/state code.

Arguments Passed :

ADDR	Integer*4	----	The site numeric code.
COD	Character*1	----	The site letter i.e. H = Hatchery, R = Release site, S = Stock site.

Arguments Returned :

ERR	Integer*4	----	The error code.
STR1	Character*20	----	The site name associated with the site code.
STR2	Character*4	----	The prodarea code if the site code is R, H and S.
STR3	Character*2	----	The province/state code associated with site.

Subroutine Call Example :
```
Call CwtSitGet( ADDR,COD,ERR,STR1,STR2,STR3 )
```

Option Codes / Description :

COD = H	----	Return information associated with hatchery number.
COD = R	----	Return information associated with release site.
COD = S	----	Return information associated with stock number.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Site is not within the valid range of addresses (ADDR is not between 0 and 9999).

Special Considerations :
1. This routine uses Fortran unit numbers 77 and 96.

CWTSML

Source File Name : CWTSML.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given the weight (WT), gear code (GEAR), and dress code (DRESS); this routine will return the grade (GRADE) of the fish as 'S', 'M', 'L', or 'U'. For chinook only.

Arguments Passed :

WT	Real*4	----	Fish weight.
DRESS	Integer*4	----	Dress code of a fish.
GEAR	Integer*4	----	The numeric gear code.
OPT	Integer*4	----	The option code.

Arguments Returned :

GRADE	Character*1	----	The character grade code.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtSML( WT,DRESS,GEAR,GRADE,OPT,ERR )
```

Option Codes / Description :
OPT = 1 ---- Pass the weight, gear and dress code, grade code returned.

Error Condition Codes / Description :
ERR = 0 ---- No error.

CWTSPORTLOC

Source File Name : CWTSPORTLOC.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given sport location (SRLOC); returns statarea (STAT), sports catch region (ICR), full name (FNAME), alternate names (ALTNAMES), latitude (LAT) and longitude (LONG).

Arguments Passed :

SRLOC	Character*7	----	The fishing site.
OPT	Integer*4	----	The option code.

Arguments Returned :

STAT	Character*3	----	The statarea code.
ICR	Integer*4	----	The sports catch region.
FNAME	Character*20	----	The full sport site name.
ALTNAMES	Character*30	----	The alternate site name(s).
LAT	Character*4	----	The latitude.
LONG	Character*5	----	The longitude.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtSportLoc( SRLOC,OPT,ERR,
                  STAT,ICR,FNAME,ALTNAMES,LAT,LONG )
```

Option Codes / Description :

OPT = 1	----	Sport location is passed and statarea, catch region, full name, alternate name(s), latitude and longitude are returned.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Record is locked due to maintenance routine, nothing is returned. Try again later.
ERR = 2	----	Statarea is not in the table so catch region is undefined. Notify SSA support group.

Other Inputs :
Source File Name : SPSTATCR.TAB

Special Considerations :
1. This routine uses Fortran unit 84.

CWTSPSTCR

Source File Name : CWTSPSTCR.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given statarea (STAT); returns sports catch region (CR) or verifies that the catch region (CR) is a sport catch region.

Arguments Passed :

CR	Integer*4	----	Numeric sports catch region.
STAT	Character*3	----	Alpha statistical area code.
OPT	Integer*4	----	The option code.

Arguments Returned :

CR	Integer*4	----	Numeric sports catch region.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtSpStCr( CR,STAT,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Statarea passed, Canadian sports catch region returned.
OPT = 2	----	Catch region passed, verify it as a sports catch region.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Can not find catch region for this statarea.
ERR = 2	----	For option 2, catch region is not a sport catch region.

Other Inputs :
Source File Name : SPSTATCR.TAB

Special Considerations :
1. This routine uses Fortran unit 100 and closes it after first call.

CWTSTAGE

Source File Name : CWTSTAGE.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This subroutine verifies release stage (STAGECOD) and returns stage name (STAGENAM) and number of table entries (NENT); or if passed the number of table entry (NENT) will return release stage name (STAGENAM) and release stage code (STAGECOD).

Arguments Passed :

STAGECOD	Character*4	---- Alpha release stage code.
STAGENAM	Character*20	---- Release stage name.
NENT	Integer*4	---- Number of table entries.
OPT	Integer*4	---- The option code.

Arguments Returned :

STAGECOD	Character*4	---- Alpha release stage code.
STAGENAM	Character*20	---- Release stage name.
NENT	Integer*4	---- Number of table entries.
ERR	Integer*4	---- The error code.

Subroutine Call Example :

```
Call Cwtstage( STAGECOD,STAGENAM,NENT,OPT,ERR)
```

Option Codes / Description :

```
OPT = 1  ----  Given release code, returns stage
               name and entry number.
OPT = 2  ----  Given entry number, returns release
               stage code and name.
```

Error Condition Codes / Description :

```
ERR = 0  ----  No errors.
ERR = 1  ----  Release name not in table (NENT returned).
ERR = 2  ----  From option = 2, invalid entry number.
ERR = 3  ----  Invalid option.
```

CWTSTATCS

Source File Name : CWTSTATCS.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Retrieves salmon catch by statarea (STAT), by year (YR) in the Mark Recovery format. All catch data for a given statarea, year and period is returned by the one call in the common block STATCSCOM.FOR.

Arguments Passed :

STAT	Character*3	----	Statistical area.
YR	Integer*4	----	The year.
PERIOD	Integer*4	----	The period code (used by Catch Statistics.
OPT	Integer*4	----	The option code.

Arguments Returned :

ERROR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtStatCs (STAT,YR,PERIOD,OPT,ERR)
```

Option Codes / Description :

OPT = 1 ---- Pass statarea, year and period; returns data in the common block STATCSCOM.FOR.

Error Condition Codes / Description :

ERR = 0 ---- No errors.
ERR = 1 ---- Year is invalid.
ERR = 2 ---- Statarea is invalid.
ERR = 3 ---- Period is undefined.

Included Source Code:
CWT:[CWTSYS.CATSAM]STATCSCOM.FOR

Special Considerations:

1. Reserves units 11, 12 and 13 which are non standard MRP units.
2. Uses salmon routines GFO, CAA1, CAREC, CAGR (Wong, 1983).

CWTSTKTYP

Source File Name : CWTSTKTYP.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
If release stock type (CODE) is passed, returns name (NAME) and number of entries in table (NVAR); or returns number of entries in table (NVAR); or if passed number of entries in table (NVAR) will return stock type (CODE) and stock name (NAME).

Arguments Passed :

CODE	Character*1	----	The stock code.
NAME	Character*20	----	The stock name.
NVAR	Integer*4	----	The table entry number.
OPT	Integer*4	----	The option code.

Arguments Returned :

CODE	Character*1	----	The stock code.
NAME	Character*20	----	The stock name.
NVAR	Integer*4	----	The table entry number.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtStkTyp( CODE,NAME,NVAR,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Passes release stock type and returns stock type name and NVAR.
OPT = 2	----	Returns maximum table entries.
OPT = 3	----	Returns stock code and name given table entry.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Invalid code passed (option 1).
ERR = 2	----	Invalid NVAR passed (option 3).

CWTSUM1

Source File Name : CWTSUM1.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine is used to adjust a known recovery to reflect NO-DATA and LOST-PIN'S. Pass year (YR) and catch region (CR) to get random marine recoveries (SUMREC).

Arguments Passed :

YR	Integer*4	----	The recovery year.
CR	Integer*4	----	The catch region.
OPT	Integer*4	----	The option code.

Arguments Returned :

SUMREC	Integer*4(54,10,4)	----	The matrix of random recovery data.
LUN	Integer*4	----	Logical unit number reserved for I/O unit.
ERROR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call Cwtsum1( YR,CR,SUMREC,LUN,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Sumrec matrix is not refreshed if routine is called with same year and catch region as previous call.
OPT = 2	----	Matrix is refreshed if same year and catch region as previous call.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	No data for this catch region.
ERR = 2	----	No data for this recovery year.

Included Source Code :
Source File Name : RECSUMCOM.FOR
Disk:[Directory] : CWT:[CWTSYS.RECOVERED]

Special Considerations :
1. Option 1 cuts down on I/O in some applications.
2. User must reserve an I/O unit in LUN. User must not use this unit and LUN must not conflict with reserved CWT units.

CWTVTP

Source File Name : CWTVTP.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine passes the numeric vessel/truck/packer (vtp) code (IVTP) and returns the alpha code (AVTP), full name (DESCVTP) and number of table entries (NVTP); or the alpha code (AVTP) is passed and the numeric code (IVTP), the full name (DESCVTP) and number of table entries (NVTP) is returned; or the full name (DESCVTP) is passed and numeric code (IVTP), alpha code (AVTP) and number of table entries (NVTP) is returned.

Arguments Passed :

ITVP	Integer*4	----	The numeric vtp code.
ATVP	Character*1	----	The alpha vtp code.
DESCVTP	Character*6	----	The full description name.
NVTP	Integer*4	----	# of entries in the table.
OPT	Integer*4	----	The option code.

Arguments Returned :

ITVP	Integer*4	----	The numeric vtp code.
ATVP	Character*1	----	The alpha vtp code.
DESCVTP	Character*6	----	The full description name.
NVTP	Integer*4	----	# of entries in the table.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call CwtVtp( ITVP,AVTP,DESCVTP,NVTP,OPT,ERR)
```

Option Codes / Description :

OPT = 1 ---- Given numeric VTP code; alpha code, full name and # of entries is returned.
OPT = 2 ---- Given alpha code; numeric code, full name and # of table entries is returned.
OPT = 3 ---- Given full name; numeric code, alpha code and # of table entries is returned.

Error Condition Codes / Description :

ERR = 0 ---- No errors.
ERR = 1 ---- Error.
ERR = 2 ---- Option error.

Special Considerations :

1. Full or descriptive name must have at least six characters.

FINCHAINS

Source File Name : FINCHAINS.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine returns starting record number for a release record (HRSNUMS) when passed hatchery, release or site number (SITE).

Arguments Passed :

SITE	Integer*4	---- Hatchery, release or stock site.
OPT	Integer*4	---- The option code.

Arguments Returned :

HRSNUMS(3)	Integer*4	---- Starting record numbers for hatchery, release and stock for this site code.
ERR	Integer*4	---- The error code.

Subroutine Call Example :

```
Call FinChains(SITE,HRSNUMS,OPT,ERR)
```

Option Codes/Descriptions:

OPT = 1 ---- Pass site number; returns starting record number to begin site chaining.

Error Conditions:

ERR = 0 ---- The site number was okay.
ERR = 1 ---- Invalid site number.

Special Considerations:

1. This program uses unit 74.

FINOPMST

Source File Name : FINOPMST.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine opens the finclip release master file RELEASE.DAF and returns the number of records (NREC) in the file .

Arguments Passed :
NONE

Arguments Returned :

NREC	Integer*4	----	The number of release records in the file.

Subroutine Call Example :

```
Call Finopmst( NREC )
```

Error Condition Codes / Description :
NONE

Special Considerations :
1. Record 1 has a null tag code and contains the number of records on file (NREC). See CWTOPMST.
2. This routine uses Fortran unit number 70.

FINRELGET

Source File Name : FINRELGET.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine or Program :
Does a direct read of the finclip release master data file using the passed tagcode (TAGCODE); or a read of the finclip release master reading the record following the previous record read.

Arguments Passed :

TAGCODE	Character*8	----	The tagcode.
OPT	Integer*4	----	The option code.

Arguments Returned :

ERR	Integer*4	----	The error code.

Subroutine Call Example :
Call Finrelget(TAGCODE,OPT,ERR)

Option Codes / Description :

OPT = 1	----	Given tagcode (TAGCODE), reads corresponding release record and puts the record into release master common blocks.
OPT = 2	----	Read record following previous record read. (File is in tagcode sequence).

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Tagcode not found.
ERR = 2	----	End of file, no record read (from option 2).

Included Common Blocks :
Source File Name : FINRELCOM.FOR
Disk:[Directory] : CWT:[FINCLIP.RELEASE]

Other Subroutines Called :
Source File Name : FINOPMST.FOR
Source File Name : FINRELRD.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]

Special Considerations :
1. This routine uses Fortran unit numbers 70 & 75.
2. Term "TAGCODE" refers to release identifier.

FINRELRD

Source File Name : FINRELRD.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine or Program :
This routine reads record from the release master data file and puts it into the release master common blocks.

Arguments Passed :
RECNUM Integer*4 ---- Record number of the release master file.

Arguments Returned :
NONE

Subroutine Call Example :
```
Call Finrelrd( RECNUM )
```

Option Codes / Description :
NONE

Error Condition Codes / Description :
NONE

Included Common Blocks :
Source File Name : FINRELCOM.FOR
Disk:[Directory] : CWT:[FINCLIP.RELEASE]

Special Considerations :
1. This routine uses Fortran unit number 70 (internal routine only).
2. File must be opened by the subroutine FINOPMST before calling FINRELRD.

FINTOCWT

Source File Name : FINTOCWT.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine moves finclip release common block variables into corresponding CWT release common block variables, allowing programs to use the same code to process both CWT and finclip release information.

Example :

```
      PROGRAM EXAMPLE
C     DO A SEQUENTIAL PASS OF BOTH RELEASE MASTERS....

      INCLUDE 'CWT:[CWTSYS.RELEASE]RELRECCOM.FOR/LIST'
      INCLUDE 'CWT:[FINCLIP.RELEASE]FINRELCOM.FOR'
C
      DO 12 II=1,2
C
C     FIRST THIS CALL IS DONE TO OPEN THE FILE AND FIND OUT
C     HOW  MANY RELEASES THERE ARE...
C
       IF(II.EQ.1)   CALL CWTOPMST(NREC)
       IF(II.EQ.2)   CALL FINOPMST(NREC)
C
C     RECORD 1 IS A CONTROL RECORD.RELEASES LIVE IN
C     RECORDS 2 THRU NREC
C
      DO 10 I=2,NREC
C
C     GET A RELEASE....
C
      IF(II.EQ.1)   CALL CWTRELREADER(I)  !PASS 1, GET A CWT
      IF(II.EQ.2) THEN       !OR, PASS 2, GET A FINCLIP,
       CALL FINRELREADER(I) !LOAD FIN COMMON,THEN
       CALL FINTOCWT        !MOVE FIN DATA BLOCK TO CWT COMMON BLOCK
      END IF

      IF(IVAR(01).NE.115)GO TO 10   !WE JUST WANT COHO...
      ETC....
```

Included Common Blocks :

Source File Name	: FINRELCOM.FOR
Disk:[Directory]	: CWT:[FINCLIP.RELEASE]
Source File Name	: RELRECCOM.FOR
Disk:[Directory]	: CWT:[CWTSYS.RELEASE]

MICROTRAN

Source File Name : MICROTRAN.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given agency micro-code (MU); returns the unique agency name code (AGENCY), coordinator code (COORD), and the country code (CNDUSA): or given agency name code (AGENCY); returns the country code (CNDUSA).

Arguments Passed :

MU	Character*2	----	The agency alpha code (1st 2 characters of tagcode).
AGENCY	Character*4	----	The agency alpha name.
OPT	Integer*4	----	The option code.

Arguments Returned :

COORD	Integer*4	----	Coordinator numeric code.
CNDUSA	Character*3	----	The country (CND or USA) alpha code.
ERR	Integer*4	----	The error code.

Subroutine Call Example :
```
Call Microtran( MU,AGENCY,COORD,CNDUSA,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Given agency micro-code; returns the agency name code, coordinator code, and the country code.
OPT = 2	----	Given agency name code; returns the country code.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	The passed agency micro-code is not defined.
ERR = 2	----	The passed agency name code is not known.

Special Considerations :
1. Some agency codes can be used by more that one agency (i.e. 62 is used by multi-American agencies).

REGIONGR

Source File Name : REGIONGR.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine validates a catch region (REG)/gear code (GR) combination.

Arguments Passed :

REG	Integer*4	----	The numeric catch region code.
GR	Integer*4	----	The numeric gear code.
OPT	Integer*4	----	The option code.

Arguments Returned :

ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call RegionGr( REG,GR,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Old gear code passed.
OPT = 2	----	New gear code passed.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Invalid catch region.
ERR = 2	----	Valid gear code not allowed for this catch region.
ERR = 3	----	Invalid option code.
ERR = 4	----	Invalid gear code.

RELSTAGE

Source File Name : RELSTAGE.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Returns the release stage name (STAGENAME) given stage code (STAGE) or vice versa.

Arguments Passed :

STAGE	Character*4	----	The release stage alpha code.
STAGENAME	Character*15	----	The release stage name.
OPT	Integer*4	----	The option code.

Arguments Returned :

STAGE	Character*4	----	The release stage alpha code.
STAGENAME	Character*15	----	The release stage name.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call RelStage( STAGE,STAGENAME,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Stage code passed, stage name returned.
OPT = 2	----	Stage name passed, stage code returned.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Stage name or stage code not found in the table.
ERR = 2	----	Stage name argument is too small to hold the stage code name, needs 15 characters.
ERR = 3	----	Stage code argument too small to hold stage code, needs 4 characters.
ERR = 4	----	Invalid option code.

Special Considerations :

1. This routine is obsolete and unsupported. It is included as an aid for program conversion to use routine CWTSTAGE.

RUNTYPE

Source File Name : RUNTYPE.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Returns the run name (NAME) given the run number (RUN); or returns the run number (RUN) given the run name (NAME); or returns successive run numbers and names.

Arguments Passed or Received :

RUN	Integer*4	----	The run number.
NAME	Character*10	----	The run name.
OPT	Integer*4	----	The option code.

Arguments Passed or Received :

RUN	Integer*4	----	The run number.
NAME	Character*10	----	The run name.
ERR	Integer*4	----	The error CODE.

Subroutine Call Example :

```
Call RunType( RUN,NAME,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Given the run number, returns the run name.
OPT = 2	----	Given the run name, returns the run number.
OPT = 3	----	Returns successive run numbers and names.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Invalid run name or run number.
ERR = 2	----	Invalid option.
ERR = 5	----	No more run numbers or names, the preceding record was the last one.

SAMLOC

Source File Name : SAMLOC.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine will verify the given sample location number (LOCN); or return the sample location name (NAME) given the location number (LOCN); or return the sample location number (LOCN) given the location name (NAME). Canadian data only.

Arguments Passed :

LOCN	Integer*4	----	The sample location number.
NAME	Character*27	----	The sample location name.
OPT	Integer*4	----	The option code.

Arguments Returned :

ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call SamLoc( LOCN,NAME,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Verify the location number.
OPT = 2	----	Given the location number, return the location name.
OPT = 3	----	Given the location name, return the location number.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Invalid location number.
ERR = 2	----	Invalid location name.

SPECIES

Source File Name : SPECIES.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Given the species Hart code (Hart) or the species Canadian code (CDNCODE); returns the species name (NAME) or vice versa.

Arguments Passed :

Hart	Integer*4	----	The numeric species Hart code.
CDNCODE	Integer*4	----	The numeric Canadian species code.
NAME	Character*10	----	The species name.
OPT	Integer*4	----	The option code.

Arguments Returned :

Hart	Integer*4	----	The numeric species Hart code.
CDNCODE	Integer*4	----	The numeric Canadian species code.
NAME	Character*10	----	The species name.
ERR	Integer*4	----	The error result code.

Subroutine Call Example :

```
Call Species( Hart,CDNCODE,NAME,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Given Hart code or Canadian code, return species name and the numeric codes. The Canadian code must be passed zero if Hart code to be used for look up.
OPT = 2	----	Given the species name, return the Hart code and Canadian code.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Invalid species Hart code or Canadian code or name.
ERR = 2	----	Invalid option (OPT).

STATSUB

Source File Name : STATSUB.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
This routine will verify catch region (CR); or verify statarea (STAT); or verify subarea (SUB); or verify catch region (CR)/statarea (STAT) combination; or verify statarea (STAT)/subarea (SUB) combination; or return the name (NAME) of the catch region (CR)/statarea (STAT)/subarea (SUB) combination; or will verify the catch region(CR)/statarea (STAT)/subarea (SUB) combination.

Arguments Passed :

CR	Integer*4	----	The numeric catch region code.
STAT	Character*3	----	The statarea code (alpha/numeric).
SUB	Integer*4	----	The numeric subarea code.
NAMLEN	Integer*4	----	The length of the above name.
OPT	Integer*4	----	The option code.

Arguments Returned :

NAME	Character*255	----	Description of the stat-subarea.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call StatSub( CR,STAT,SUB,NAME,NAMLEN,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Verifies the given catch region.
OPT = 2	----	Verifies the given statarea.
OPT = 3	----	Verifies the given subarea.
OPT = 4	----	Verifies the given catch region/statarea.
OPT = 5	----	Verifies the given statarea/subarea.
OPT = 6	----	Returns name of the catch region/statarea/subarea.
OPT = 7	----	Verifies the catch region/statarea/subarea.

Error Condition Codes / Description :

ERR = 0	----	No errors.
ERR = 1	----	Invalid catch region.
ERR = 2	----	Invalid statarea

ERR = 3	----	Invalid subarea.
ERR = 4	----	Invalid catch region/statarea combination.
ERR = 5	----	Invalid statarea/subarea combination.
ERR = 6	----	Cannot find catch region/statarea/subarea.
ERR = 7	----	Name argument not long enough, actual name truncated.
ERR = 8	----	Invalid option.

Special Considerations :

1. This routine uses Fortran unit number 91 (internal routine only).
2. If user program not linked to latest version of the table file. First call to statsub will abort program with message for user to re-link.
3. For option 7, if subarea is 0 then verify catch region/statarea. If subarea is 0 and statarea is blank or '000' then verify the catch region.

STUDYTYPE

Source File Name : STUDYTYPE.FOR
Disk:[Directory] : CWT:[CWTSYS.TABLES]
Program Language : FORTRAN

Purpose of Subroutine :
Verifies the given study type code (CODE) and returns the study name (NAME).

Arguments Passed :

CODE	Character*1	----	The study type alpha code.
OPT	Integer*4	----	The option code.

Arguments Returned :

NAME	Character*20	----	The study type name.
ERR	Integer*4	----	The error code.

Subroutine Call Example :

```
Call StudyType( CODE,NAME,OPT,ERR )
```

Option Codes / Description :

OPT = 1	----	Given study type code, return the study type name.

Error Condition Codes / Description :

ERR = 0	----	No error.
ERR = 1	----	Invalid study type code.

APPENDIX A

The subroutine **MICROTRAN** is a table carrying agency codes and the country they are affiliated with. Agencies are individual government departments in both the United States and Canada that participate in marking salmon. Agencies are assigned individual numeric codes, i.e. Canada uses agency codes 02, 08 and 12 and these codes are incorporated into the binary coding used in tagging. Each agency is responsible for reporting tagcodes used to PSMFC. The first two characters in the following table are the agency code and this code can be 'owned' by a group or is some cases one code will have multiple agency use. The next three characters indicate the country in which the agency exists. The last two characters are the coordinator code and indicate to which coordinator that agency will report the use of any tagcodes. For example, agency code 02 is used by the Canadian Department of Fisheries and Oceans in Canada (CDFO) and CDFO reports all tagging information to coordinator 03. A list of all agencies and coordinators can be found in the RMPC report on salmonid releases.

```
01WDF USA04
02CDFOCDN03
03NMFSUSA02
04ADFGUSA01
05FWS USA07
08CDFRCDN03
10IDFGUSA10
12BCFWCDN09
12BCEPCDN09
  HFADFGUSA
  ANADUSA05
13WDF USA04
14FWS USA07
15WDF USA04
21NIFCUSA14
23NMFSUSA06
31ADFGUSA01
47MIC USA13
60OAF USA05
63WDF USA04
B6ADFGUSA
H1WDF USA04
H7ODFWUSA05
[illegible]NMFSUSA06
[illegible]NMFSUSA06
```

APPENDIX B

Statweeks are labelled by a three digit code. The first two digits delineate the month and the last one specifies the week within the month. Each statweek in the catch database has 7 days and each month has 4 statweeks except that months 04, 07 and 10 each have 5 periods. Statweek 120 as all periods from December through February, inclusive.

The table below for 1975 gives the week ending day and the statweek (MMW). A table for any year can be generated by running:

$run ssa:[CWTSYS.TABLES]MMWTAB

ENDING DAY	MON	YR	STATWEEK	PERIOD
08	03	75	031	1
15	03	75	032	2
22	03	75	033	3
29	03	75	034	4
05	04	75	041	5
12	04	75	042	6
19	04	75	043	7
26	04	75	044	8
03	05	75	045	9
10	05	75	051	10
17	05	75	052	11
24	05	75	053	12
31	05	75	054	13
07	06	75	061	14
14	06	75	062	15
21	06	75	063	16
28	06	75	064	17
05	07	75	071	18
12	07	75	072	19
19	07	75	073	20
26	07	75	074	21
02	08	75	075	22
09	08	75	081	23
16	08	75	082	24
23	08	75	083	25
30	08	75	084	26
06	09	75	091	27

13	09	75	092	28
20	09	75	093	29
27	09	75	094	30
04	10	75	101	31
11	10	75	102	32
18	10	75	103	33
25	10	75	104	34
01	11	75	105	35
08	11	75	111	36
15	11	75	112	37
22	11	75	113	38
29	11	75	114	39
DEC,JAN,FEB		75	120	40
TOTAL (ALL PERIODS)				41

APPENDIX C

The remaining appendices list example programs using various subroutines and common blocks.

Read Release Shell

```
      PROGRAM SEQREAD
      IMPLICIT INTEGER*4 (A-Z)
C
C     ILLUSTRATES HOW TO READ THE CWT RELEASE MASTER
C     IN A SEQUENTIAL MANNER IN TAGCODE SEQUENCE
C
C     $LINK,SEQREAD,CWT:[CWTSYS.TABLES]CWTLIB/LIB
C
C     PROGRAM GETS RELEASE RECORD PASSED TO HIM IN
C     COMMON BLOCK BELOW.
      INCLUDE 'CWT:[CWTSYS.RELEASE]RELRECCOM.FOR/LIST'
C
C     FIRST THIS CALL IS DONE TO OPEN THE FILE AND FIND
C     OUT HOW  MANY RELEASES THERE ARE...
C
      CALL CWTOPMST(NREC)
C
C     RECORD 1 IS A CONTROL RECORD.RELEASES LIVE IN
C     RECORDS 2 THRU NREC
C
      DO 10 I=2,NREC
C
C     GET A RELEASE....
C
      CALL CWTRELREADER(I)
      IF(DELSW.EQ.'Y') GO TO 10  !RELEASE FLAGGED
C                                ! INVALID
C     ******************************
C     *                            *
C     * INSERT YOUR CODE HERE *
C     * (EG,SELECTION CRITERIA,    *
C     *  ACCUMULATION RTNS)        *
C     ******************************
C
10    CONTINUE
C
C     ******************************
C     *   END OF FILE PROCESSING   *
C     *        GOES HERE           *
C     ******************************
C
      END
```

APPENDIX D

Read Recovery Shell

```
      PROGRAM SHELL
      IMPLICIT INTEGER*4 (A-Z)
C
C     "SEQUENTIAL PASS" OF RECOVERY FILE FOR SOME YR...
C
C     DATA FOR INDIVIDUAL RECOVERIES PASSED VIA COMMON
C        BLOCK BELOW.
C
      INCLUDE 'CWT:[CWTSYS.RECOVERED]RECTAGCOM.FOR/LIST'

C
      DO 200 YR=Y1,Y2    !MAKE Y1,Y2 THE YEARS YOU WANT
C
C
      DO 100 I=1,999999
         OPT=1
         CALL CWTRECRD(YR,I,OPT,ERR)
         IF(ERR.NE.0) GO TO 150  !END THIS YEARS DATA
C
C*******************************************************
C        PUT YOUR SELECTION CODING,CALCULATIONS ETC IN
C              HERE
C
C*******************************************************
C
100   CONTINUE
C
150   CONTINUE  !END CURRENT YEAR
C
C*******************************************************
C     PUT YOUR YEAR END CODE HERE
C*******************************************************
C
C
200   CONTINUE  !END YEAR LOOP
C*******************************************************
C     PUT YOUR EOJ CODE HERE
C*******************************************************
900   CONTINUE
C
      STOP
      END
```

APPENDIX E

Combined Release and Catch Samples Shell

```
      PROGRAM SHELL3
      IMPLICIT INTEGER*4 (A-Z)
C
C     "SEQUENTIAL PASS" OF RECOVERY FILE FOR SOME YR...
C     ACCESS RELEASE AND CATCH/SAMPLE DATA ON SELECTED
C     RECOVERIES.
C     THIS CODE ILLUSTRATES THE USE OF SOME
C     SUBROUTINES.IT IS NOT  GIVEN AS AN EXAMPLE OF
C     THE MOST EFFICIENT ROUTE TO THE VARIOUS DATA SETS.
C
C     COMMON BLOCK FOR RECOVERY RECORDS:
      INCLUDE 'CWT:[CWTSYS.RECOVERED]RECTAGCOM.FOR/LIST'
C
C     COMMON BLOCK FOR RELEASE RECORDS:
      INCLUDE 'CWT:[CWTSYS.RELEASE]RELRECCOM.FOR/LIST'
C
C     DECLARATION FOR CATCH/SAMPLE DATA:
C     (THE C/S DATA IS A PASSED ARGUMENT,NOT DONE VIA
C      COMMON)
      INCLUDE 'CWT:[CWTSYS.CATSAM]CSRECCOM.FOR/LIST'
C
      DO 200 YR=Y1,Y2    !MAKE Y1,Y2 THE YEARS U WANT
C
C
      DO 100 I=1,999999  !"LOOP DRIVER"  LOOP WILL EXIT
C                                !   EARLIER
         OPT=1
         CALL CWTRECRD(YR,I,OPT,ERR)
         IF(ERR.NE.0) GO TO 150  !END THIS YEARS DATA
C
C*************************************************
C        PUT YOUR SELECTION CODING,CALCULATIONS ETC IN
C        HERE
C        PRETEND WE WANT EXPANDABLE MARINE RECOVERIES OF
C        CWT RELEASES.
C
      IF(RECFORMAT.NE.1) GO TO 100  !WANT ONLY MARINE
C                                   !RECOVERIES
C
C     LOOK SEE IF IT'S A "NON-TAG".(EG NO-PIN, LOST-PIN)
C
      NOPT=3
      CALL CWTNONTAG(RECTAG,DUMMY,NOPT,NERR) !SEE
C                                            !CWTNONTAG DOC
      IF(NERR.EQ.0) THEN   !IT IS A NON-TAG
         GO TO 100
      END IF
```

```
C
C
C     GET CATCH/SAMPLE DATA.
C
      CSOPT=1
      I4REGION=RECREGION  !SOME VARIABLES HAVE DIFFERENT
C                         !TYPES
C                         !SO - RECREGION IS I*2,RTN
C                         !WANTS I*4
      CALL CWTCSGET(I4REGION,YR,CSREC,CSOPT,CSERR)
      IF(CSERR.NE.0) THEN
        GO TO 100         !NO CATCH/SAMPLE AVAILABLE
      END IF
C
C     GET RELEASE DATA.
C
      ROPT=1   !HIT FILE BY TAGCODE
      CALL CWTRELGET(RECTAG,ROPT,RERR) !LOAD RELEASE
C                                      !DATA INTO COMMON
C
C
C   AT THIS POINT,PROGRAM HAS AVAILABLE A MARINE
C   RECOVERY,THE RELEASE
C   RECORD FOR THIS TAGCODE,AND THE CATCH/SAMPLE DATA
C   FOR THE CATCH
C   REGION AND YEAR THE RECOVERY OCCURRED IN.
C
C*************************************************
100   CONTINUE
C
150   CONTINUE  !END CURRENT YEAR
C
C*************************************************
C     PUT YOUR YEAR END CODE HERE
C*************************************************
C
200   CONTINUE  !END YEAR LOOP
C
***************************************************
C     PUT YOUR EOJ CODE HERE
C*************************************************
900   CONTINUE
C
      STOP
      END
```

REFERENCES

Holmes, M.A. and D.W.A. Whitfield. 1991. User's Manual for the Commercial Catch Spreadsheet Program. Can. Tech. Rep. Fish. Aquat. Sci. 1807 : 44 p.

Kuhn, B.R. 1988. The MRP-Reporter program: A data extraction and reporting tool for the mark recovery program database. Can. Tech. Rep. Fish. Aquat. Sci. 1625: 145 p.

Northwest Marine Technology, Inc. 1990. Sequentially Marked Coded-wire Tags. Dept. of Fisheries & Oceans in-house document.

Regional Mark Processing Center Pacific States Marine Fisheries Commission: Pacific Salmonid Coded Wire Tag Releases through 1990. Manual: 418 p.

Serbic, Greg. 1991. The Salmon Escapement Database and Reporting System. Can. Tech. Rep. Fish. Aquat. sci. 1791 : 104 p.

Whitehead, Valerie. 1990. Commercial Catch Statistics System, Workflow, Procedures, and Data. Dept. of Fisheries & Oceans in-house document.

Wong, F.Y.C. 1983. Historical salmon commercial catch data system of the Fisheries Research Branch, Department of Fisheries and Oceans, Pacific Region. Canadian Technical Report of Fisheries and Aquatic Sciences No. 1156 : 94 p.